intel

**APPLICATION NOTE**

**AP-102**

April 1981

# 2816 Microprocessor Interface Considerations

**John F. Rizzo**
Special Products Division
Applications Engineering



AFN-01913A

## INTRODUCTION

$E^2$—Electrically Erasable, that's the key to the new 2816. The flexibility of RAM and the non-volatility of ROM have now been merged to form $E^2$. System designers can now benefit from in-circuit changes to non-volatile program and data storage. Microprocessor-based systems can be extended to a higher level of functionality and performance, while costs associated with software changes, maintenance and service can be dramatically reduced. A ROM with RAM-like flexibility—that's $E^2$.

This application note will discuss the concept of microprocessor interface to the 2816. Because $E^2$ encompasses both RAM and ROM, the interface concepts are unique. In this note, the control interface will be discussed specifically (four of which are detailed here). The concept of $V_{PP}$ switching, and chip erase control circuits are also presented. Finally, using multiple 2816's in-system will be shown. In previous application notes (AP-101) the component characteristics were discussed. Here we will detail the interface of the component to the processor.

The specifications of the 2816 have been discussed in detail in AP-101. The most unique characteristic of the interface with the microprocessor is the concept of the write access. The read operation is fairly straightforward in that it does not depart from traditional EPROM concepts. The read operation is very fast, allowing compatibility with current and future microprocessors, benefiting the user with highest possible throughput and system performance. Because the write cycle time is not the same as read access, a unique situation exists for the system designer.

Because the 2816 requires a write time of approximately 10 milliseconds, there is an intrinsic timing difference between the microprocessor and the memory. If one applied the 10 millisecond write time to the write cycle time of the microprocessor, one could execute approximately 50 thousand write cycles in the duration of 10 milliseconds. Additional circuitry is required to properly interface these timing differences. There are several approaches for doing this, several of which will be discussed.

## BUS INDEPENDENT TRANSFER

These approaches can be broken down into two general categories: bus dependent and bus independent. The bus independent concept allows the microprocessor to run at full speed while the 2816 write operation progresses. The microprocessor sends out a write operation just as usual, except that a control interface continues the 10ms write cycle independent of the CPU. The microprocessor is notified at some later time that the write operation is finished. This can occur either through interrupt service, or through an I/O polling operation. Thus, the microprocessor can run independently of the $E^2$ controller during the write time. Appropriately, it is "bus independent." Table 1 shows a partial list of appropriate applications using this controller type.

**Table 1. Bus Independent Applications**

| |
|---|
| CRT Terminal Control |
| Navigation Computers |
| Industrial Controllers |
| Telecommunications |
| Military Computers |

## BUS DEPENDENT TRANSFER

The other approach involves dedicating the microprocessor during the $E^2$ write cycle. In this case wait states are inserted into the memory cycle as the write is proceeding. The disadvantage of such an approach is that the microprocessor is inhibited from doing any other operation during the 10 millisecond write time.

In many applications, however, this can be a suitable solution to the 2816 control issue. An example is the case where information is transferred into the $E^2$ on system power up or power down. During the power sequencing times, one expects that the system would not be executing any other instructions, or in fact, doing anything other than servicing the $E^2$ device. In terms of hardware, this scheme would be implemented by controlling the microprocessor's ready or wait line while the write is occurring. This approach offers the advantage of being very simple to implement and does not require any software overhead in terms of interrupt service or I/O polling. Additionally, this scheme is acceptable in many applications where erase/write is only occasional. Such an interface is termed bus dependent. Table 2 provides an applications guide for this interface.

**Table 2. Bus Dependent Applications**

| |
|---|
| Program Storage |
| Look-up Tables |
| Remote Data Collection |

We will show that the two distinct control applications dictate the amount of hardware required to interface the device to the microprocessor, as well as the efficiency at which the information transfer occurs. Above all, the individual application area for the $E^2$ will uniquely determine the kind of control circuitry that is required.

Based on these two distinct areas, we will discuss several different recommended interfaces that have been generated for use with the device. Though these controllers were designed to operate in an 8085/8088/8086 based system, they can be easily adapted to any kind of microprocessing environment.

## INTERFACE OVERVIEW

There are five controllers at present, four of which are available for use with the 2816 Demonstration Unit. The Controller I is a small scale integration implementation which uses the microprocessor's ready line as a means of inserting wait states into the memory cycle. It is a very simple controller application; one that is dedicated to the microprocessor. For this controller, the microprocessor is inhibited from operating during the time that the 2816 is being written to. Figure 1 is a block diagram for this control interface.

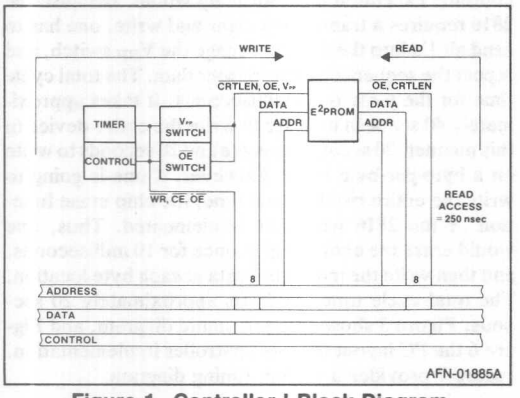



**Figure 1. Controller I Block Diagram**

The Controller II implementation is an interrupt driven interface, which requires little software overhead. In this case, the information is sent into the interface while the microprocessor simply strobes the write line as normal. The controller then handles all the necessary latching and generation of signals for the $E^2$ device. At the completion of the write cycle, the controller signals the microprocessor with a restart vector to interrupt service routines. The block diagram for Controller II is shown in Figure 2.

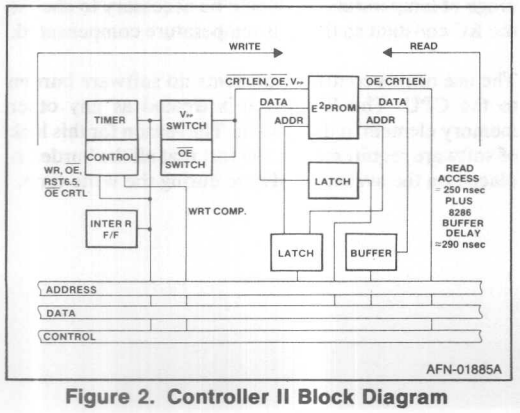



**Figure 2. Controller II Block Diagram**

The Controller III design is a more integrated version of II; it uses an Intel 8155 for controlling, latching, timing, and other functions. This controller, however, requires software in order to drive the 8155 and to set up the proper address/data lines to the 2816 during the write cycle. See Figure 3 for this block diagram.

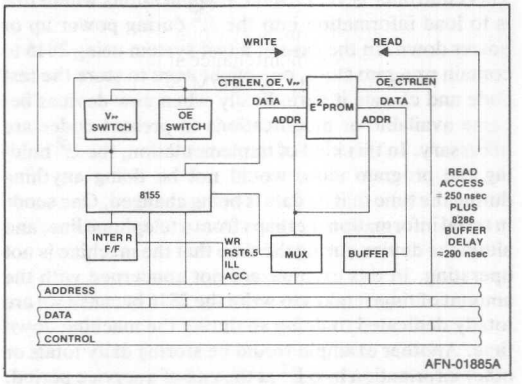



**Figure 3. Controller III Block Diagram**

The Controller IV implementation is a more highly integrated version of III; it uses an 8155 for writing and reading of the 2816. It also requires more software for the necessary initializations. A block diagram is given in Figure 4. Controllers I through III allow the 2816 to be read at very high speeds. Controller IV, however, requires long read times as reading occurs through the 8155 I/O port.

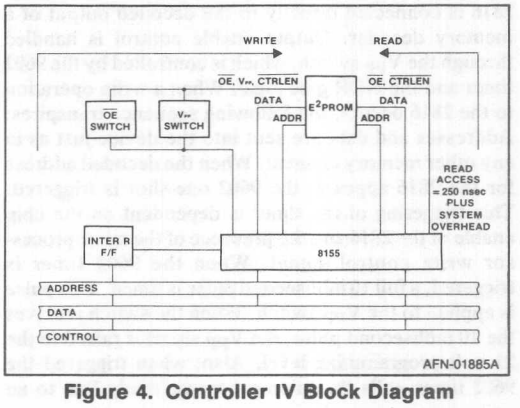



**Figure 4. Controller IV Block Diagram**

Controller V is an interface using a Bipolar PROM as a state machine. In this case there are two separate addresses for the $E^2$ device in the system; each of which corresponds to a different controller function. The first address corresponds to reading and writing of the $E^2$, the second address to chip erasure of the 2816. This controller is easily applied where a large memory space is available, as in a 16-bit microprocessing system.

## CONTROLLER I DESCRIPTION

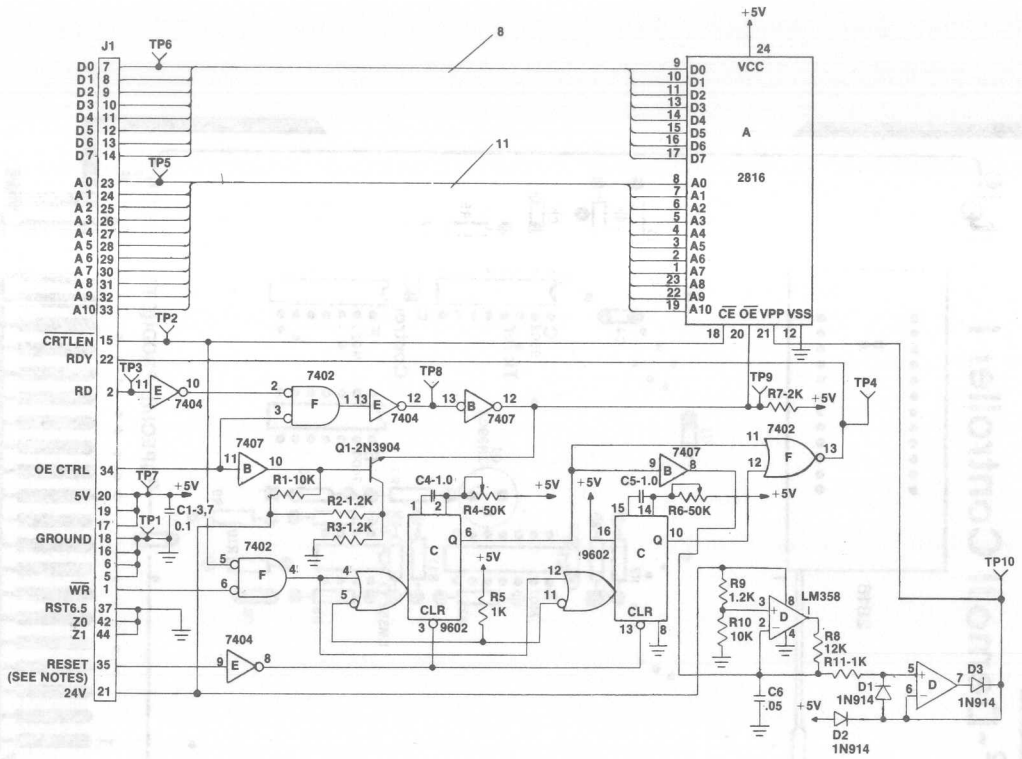
Examining the controller implementations in more detail, we find that the Controller I interface inhibits the microprocessor from operating during the write time. This controller is very useful in applications where one is to load information into the $E^2$ during power up or power down. In the case of a test system using 2816 to contain program store, one might want to store the test code and change it periodically when new devices become available or modifications to present codes are necessary. In this kind of implementation, the $E^2$ holding the program store would not be doing anything during the time that its data is being changed. One sends in serial information, perhaps from a telephone line, and alters the device during the time that the machine is not operating. In this case we are not concerned with the amount of time it takes to write the 2816 because we are totally dedicated to doing so during the machine down time. Another example would be storing daily totals or other information into $E^2$ at the end of a service period. In this case, when the machine is powered down it will automatically update the 2816 as a data memory. The amount of time it takes to do this is irrelevant because the machine is totally dedicated to the task during its shut down period.

The Controller I implementation discussed here uses three components in the system, shown in Figure 1. The 2816 address and data lines are connected directly onto the microprocessor bus. The chip enable line for the 2816 is connected directly to the decoded output of a memory decoder. Output enable control is handled through the $V_{PP}$ switch, which is controlled by the 9602 timer and the NOR gate logic. When a write operation to the 2816 occurs, the following sequence transpires: Addresses and data are sent into the device just as in any other memory element. When the decoded address for the 2816 appears, the 9602 one-shot is triggered. This triggering of the timer is dependent on the chip enable of the 2816 and the presence of the microprocessor write control signal. When the 9602 timer is triggered, a full 10 millisecond pulse is timed. This pulse is applied to the $V_{PP}$ switch. When the switch receives the 10 millisecond pulse, the $V_{PP}$ signal is raised to the 21 volt programming level. Also, when triggered the 9602 timer pulls the microprocessor ready line to an inactive low level. This signals the microprocessor that the memory element is not ready to relinquish the data bus, or indeed requires a long write time.

The ready line inhibits the microprocessor from incrementing the program counter and causes the processor to provide stable signals to the 2816 during the 10ms pulse. At the completion of 10ms, the timer disengages the $V_{PP}$ switch, stopping the write. It also pulls the microprocessor ready line to high level. When the ready line is pulled high, it indicates that the memory element has completed its cycle and that the microprocessor can continue execution as it normally would. Because the 2816 requires a transparent clear and write, one has to send all 1's into the device, engage the $V_{PP}$ switch, and repeat the sequence for the proper data. The total cycle time for the write is 20 milliseconds. It takes approximately 40 seconds in order to write the entire device in this manner, 20 seconds to erase and 20 seconds to write on a byte-per-byte basis. However, if one is going to write the entire block at one time, the chip erase function of the 2816 would be implemented. Thus, one would erase the entire chip at once for 10 milliseconds, and then write the individual data at each byte location. The total cycle time would be approximately 20 seconds. Figure 5 shows the schematic diagram, and Figure 6 the PC layout for this controller implementation. Figure 7 provides a system timing diagram.

The components mentioned were chosen for Controller I more for convenience than for circuit design requirements. Conceivably, one could have other devices operating in the system to provide timing of the 10 millisecond pulse and switching of the $V_{PP}$ signals. A programmable timer could exist within the microprocessing environment and could time out the 10 milliseconds more accurately than is possible with the 9602. One of the difficulties with the one-shot is the inherent variability of the RC time constant used to time 10 milliseconds. If the system is to operate over a wide range of temperatures, it would be necessary to choose the RC constant so that it is temperature compensated.

The use of this controller presents no software burden to the CPU. The $E^2$ device is treated as any other memory element in the system. The reason for this lack of software requirement is the fact that all the burden is placed on the system hardware during the write time.




NOTES: (UNLESS OTHERWISE SPECIFIED)

1. RESET SIGNAL ORIGIN IS SYSTEM DEMONSTOR UNIT J1-22.
2. RESISTOR VALUES ARE IN OHMS, 1/4W, ± 5%.
3. +5V CONNECTED TO PIN 14 AND GROUND CONNECTED TO PIN 7 ON INTEGRATED CIRCUITS.
4. TEST POINTS
   1. GROUND
   2. CRTLEN
   3. READ
   4. READY
   5. ADDRESS 0
   6. DATA 0
   7. VCC
   8. OE CONTROL-READ
   9. RD
   10. VPP
5. ALL DIODES IN914.
6. ALL CAPACITORS IN $\mu$f.

AFN-01885A

**Figure 5. E²-Demo Controller I**



AFN-01913A

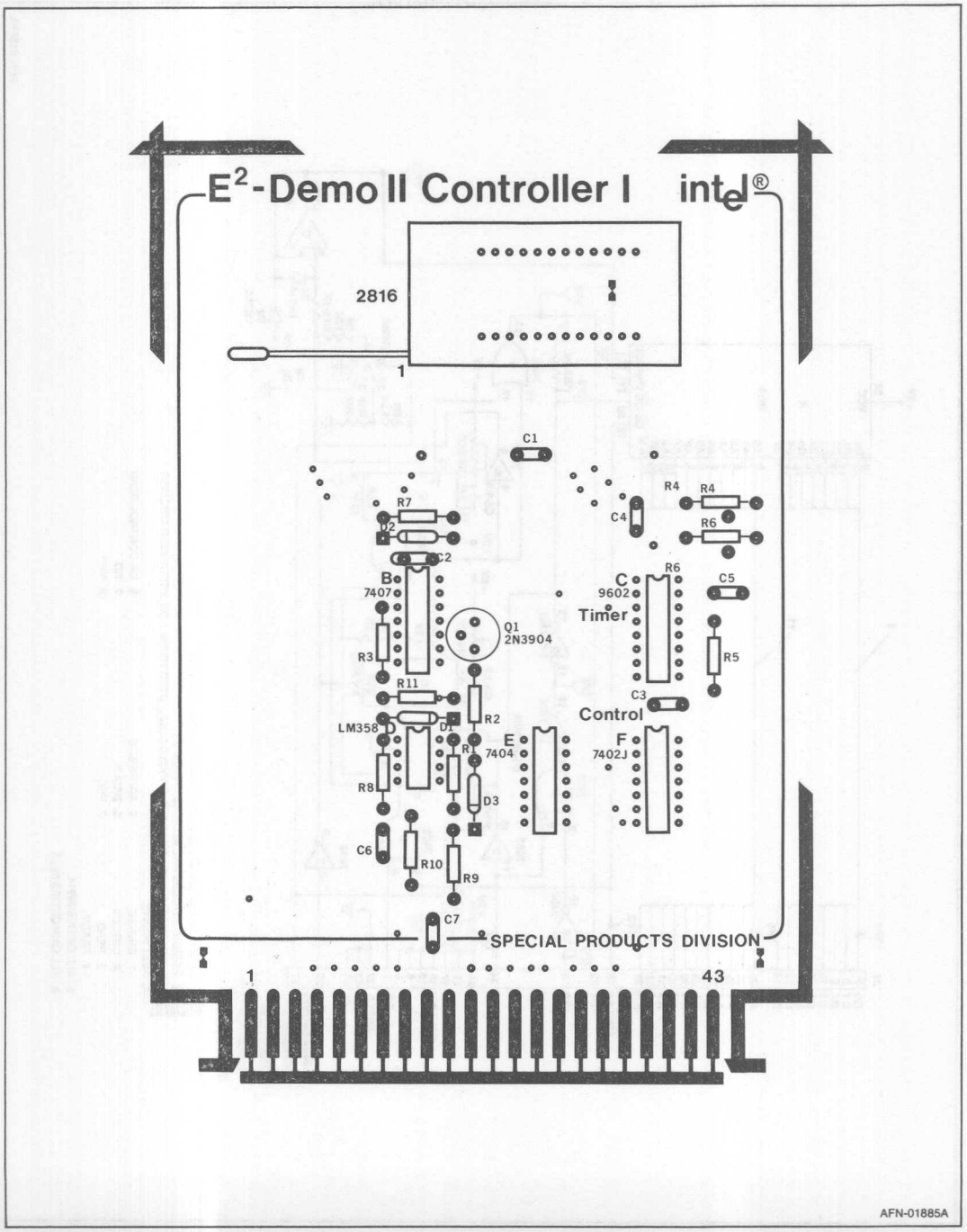



**Figure 6a. E²-Demo II Controller I**



AFN-01913A

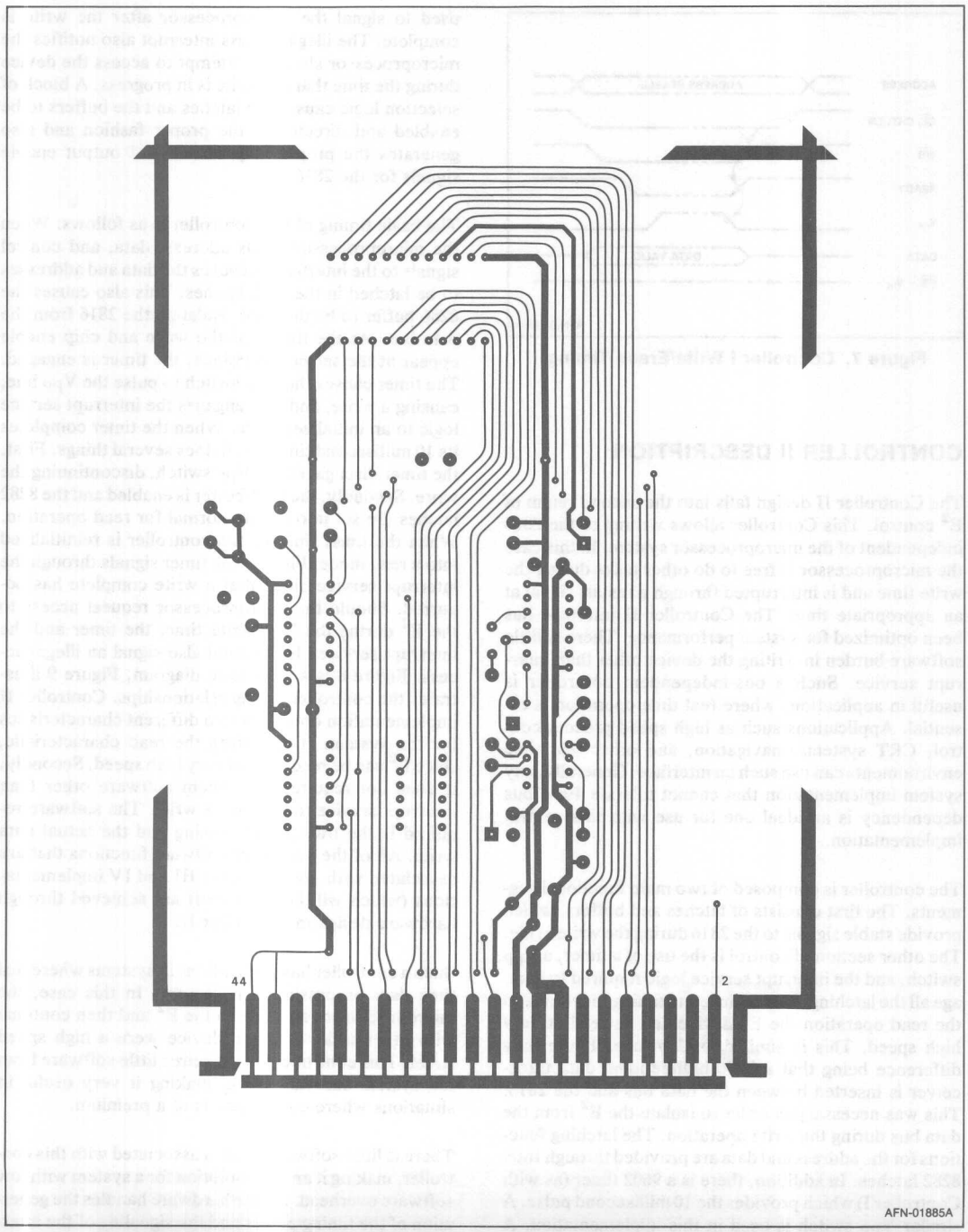



Figure 6b. $E^2$-Demo II Controller I (Continued)

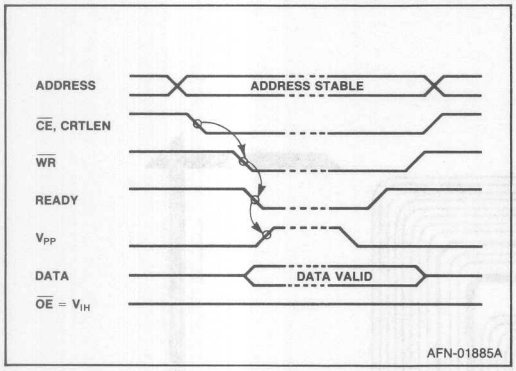



**Figure 7. Controller I Write/Erase Timing**

## CONTROLLER II DESCRIPTION

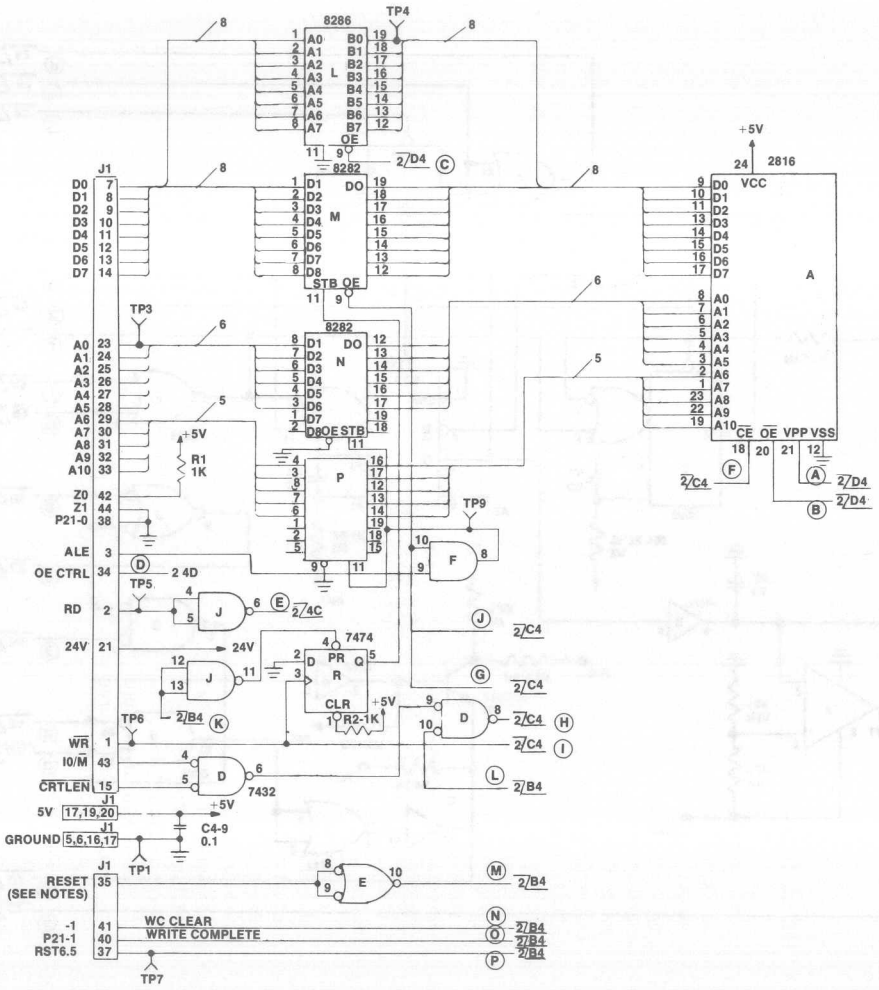
The Controller II design falls into the second realm of $E^2$ control. This Controller allows writing of the 2816 independent of the microprocessor system. In this case the microprocessor is free to do other tasks during the write time and is interrupted through a restart signal at an appropriate time. The Controller II interface has been optimized for system performance. There is little software burden in writing the device other than interrupt service. Such a bus-independent controller is useful in applications where real time operation is essential. Applications such as high speed process control, CRT systems, navigation, and other real time environments can use such an interface. Generally, any system implementation that cannot tolerate 10ms bus dependency is an ideal one for use with this control implementation.

The controller is composed of two main functional segments. The first consists of latches and buffers, which provide stable signals to the 2816 during the write cycle. The other section of control is the use of a timer, a $V_{PP}$ switch, and the interrupt service logic required to manage all the latching, controlling, and timing functions. In the read operation the $E^2$ device can be read at very high speed. This is similar to Controller I, the only difference being that a 8286 bidirectional data transceiver is inserted between the data bus and the 2816. This was necessary in order to isolate the $E^2$ from the data bus during the write operation. The latching functions for the address and data are provided through Intel 8282 latches. In addition, there is a 9602 timer (as with Controller I) which provides the 10 millisecond pulse. A similar $V_{PP}$ switch is used in this implementation. A block of interrupt service logic, which provides write complete interrupts and illegal-access interrupts, is used to signal the microprocessor after the write is complete. The illegal-access interrupt also notifies the microprocessor should it attempt to access the device during the time that the write is in progress. A block of selection logic causes the latches and the buffers to be enabled and directed in the proper fashion and also generates the proper chip enable and output enable signals for the 2816.

The basic timing of this controller is as follows: When the microprocessor sends address, data, and control signals to the interface, it causes the data and addresses to be latched in the 8282 latches. This also causes the 8286 buffer to be disabled, isolating the 2816 from the data bus. At the time that the write and chip enable appear at the select logic block, the timer is engaged. The timer causes the $V_{PP}$ switch to pulse the $V_{PP}$ line, causing a write, and also engages the interrupt service logic to an initialized state. When the timer completes its 10 millisecond time out, it does several things. First, the timer disengages the $V_{PP}$ switch, discontinuing the write. Secondly, the 8286 buffer is enabled and the 8282 latches are set into a state normal for read operation. When the timer finishes the controller is reinitialized into a read mode. Finally, the timer signals through the interrupt service block that a write complete has occurred. Should the microprocessor request access to the $E^2$ during the long write time, the timer and the interrupt service block would also signal an illegal access. Figure 8 is a schematic diagram, Figure 9 illustrates the controller timing relationships. Controller II implementation optimizes two different characteristics for the system. It optimizes the read characteristic, since $E^2$ can be read from at very high speed. Secondly, it does not require any system software other than interrupt service to perform a write. The software required is the transparent erasing and the actual data write. All of the necessary software functions that are associated with the Controller III and IV implementations (which will be discussed) are achieved through hardware design in Controller II.

Such a controller has applications in systems where real time data processing is necessary. In this case, the microprocessor can write to the $E^2$ and then continue with other tasks as if the device were a high speed RAM. This controller also requires little software from the system software bank, making it very useful in situations where code space is at a premium.

There is little software burden associated with this controller, making it an ideal solution for a system with low software overhead. All the hardware handles the generation of the timing pulses and the signaling of the interrupt service at the proper time. Figure 10 shows the printed circuit layouts.




NOTES: (UNLESS OTHERWISE SPECIFIED)

1. RESET SIGNAL ORIGIN IS SYSTEM DEMONSTRATOR UNIT JI-22.
2. RESISTOR VALUES ARE IN OHMS, 1/4W, ±5%.
3. +5V CONNECTED TO PIN 14 AND GROUND CONNECTED TO PIN 7 ON INTEGRATED CIRCUIT.
4. TEST POINTS
   1. GROUND
   2. CE
   3. ADDRESS 0
   4. DATA
   5. READ
   6. WRITE
   7. RST65
   8. WRITE PULSE
   9. ALE-WRITE LOCKOUT
5. ALL DIODES IN914.
6. ALL CAPACITORS IN $\mu$f.

**Figure 8a. E$^2$-Demo Controller II**



AFN-01913A

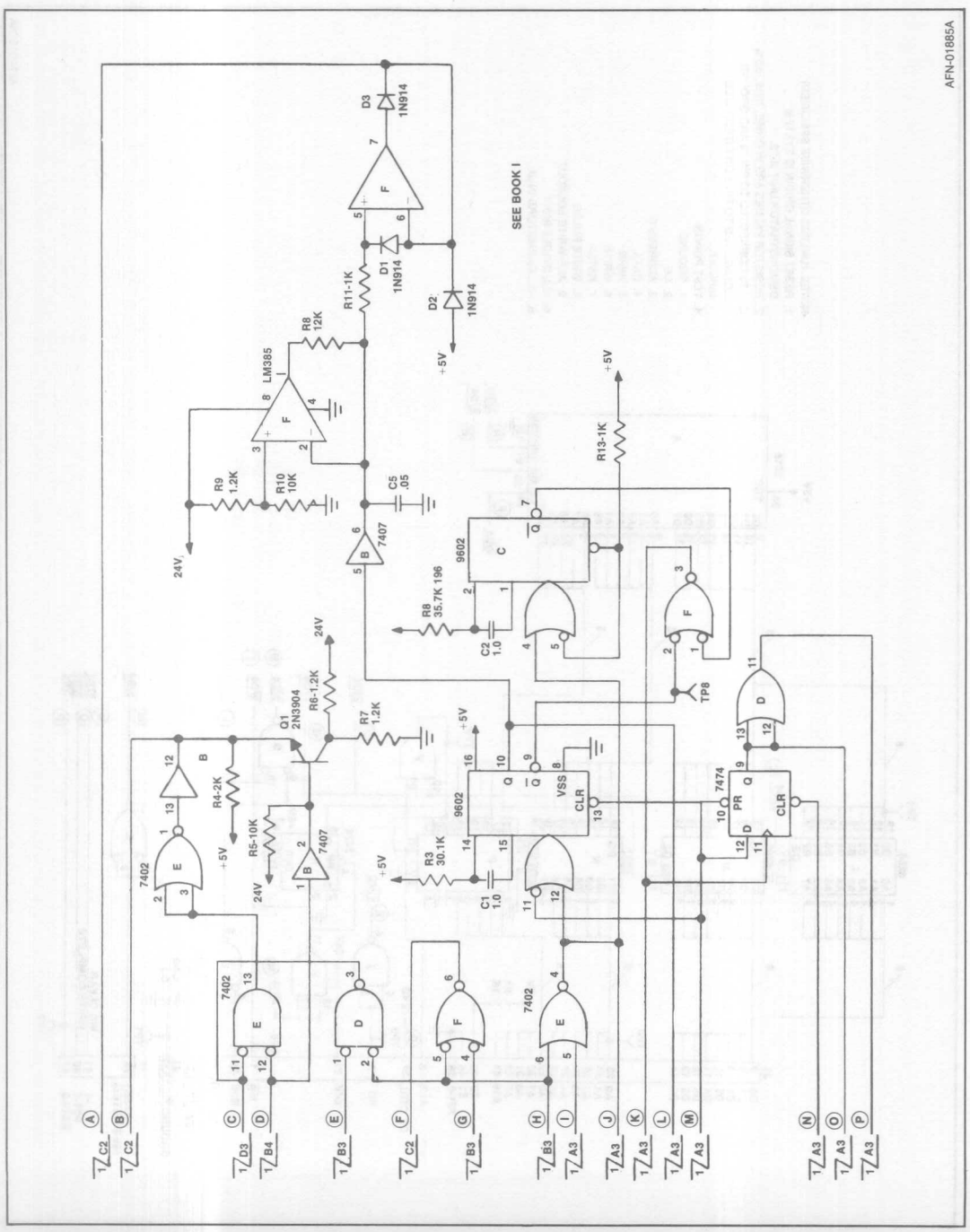



Figure 8b. $E^2$-Demo Controller II (Continued)

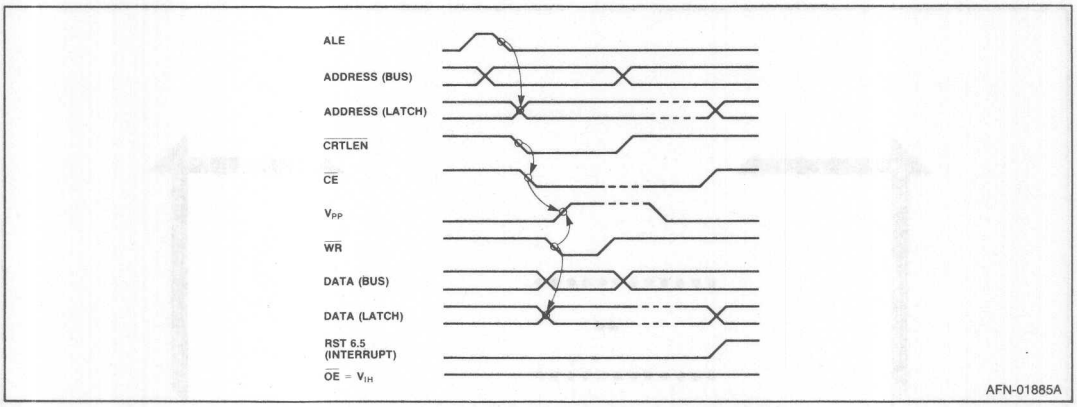



**Figure 9. Controller II Write/Erase Timing**

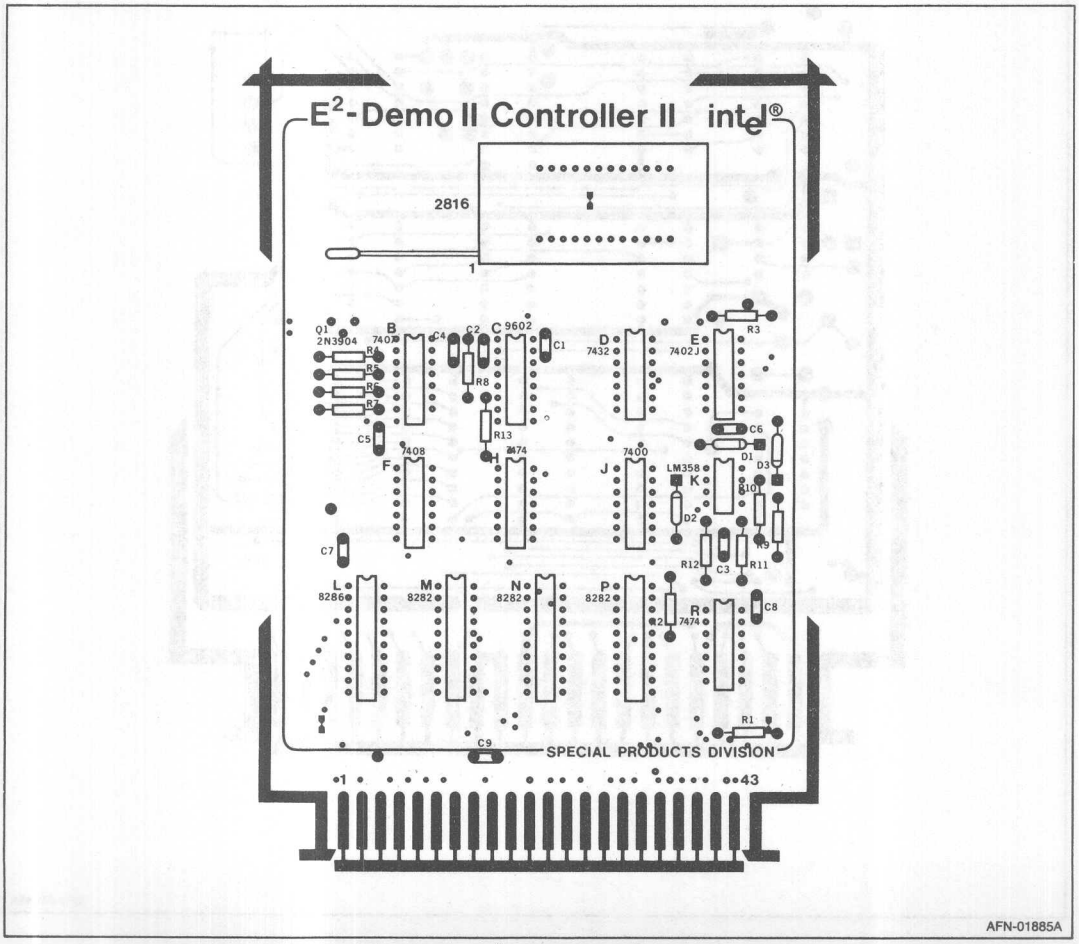



**Figure 10a. E²-Demo II Controller II**



AFN-01913A

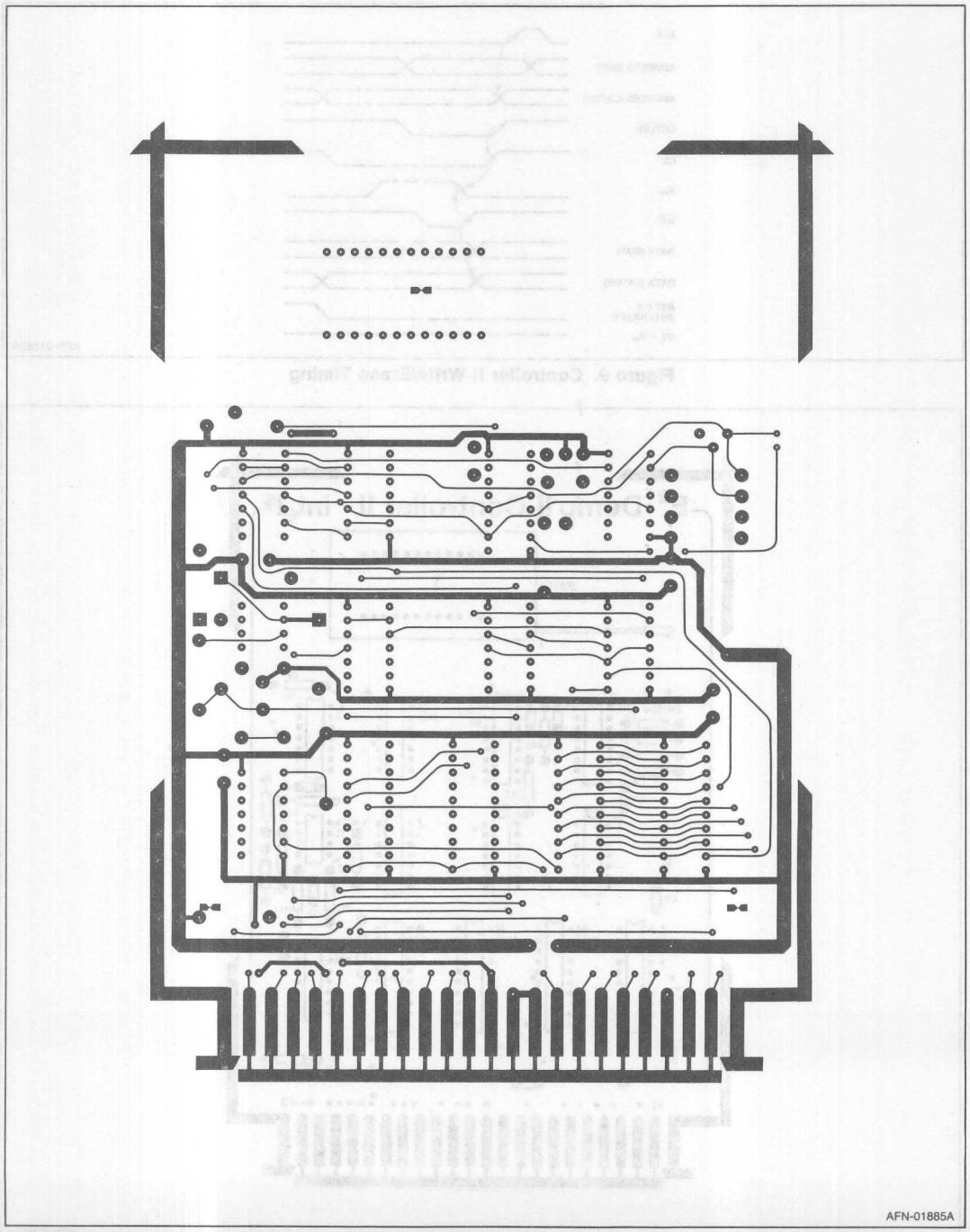



Figure 10b. $E^2$-Demo II Controller II (Continued)

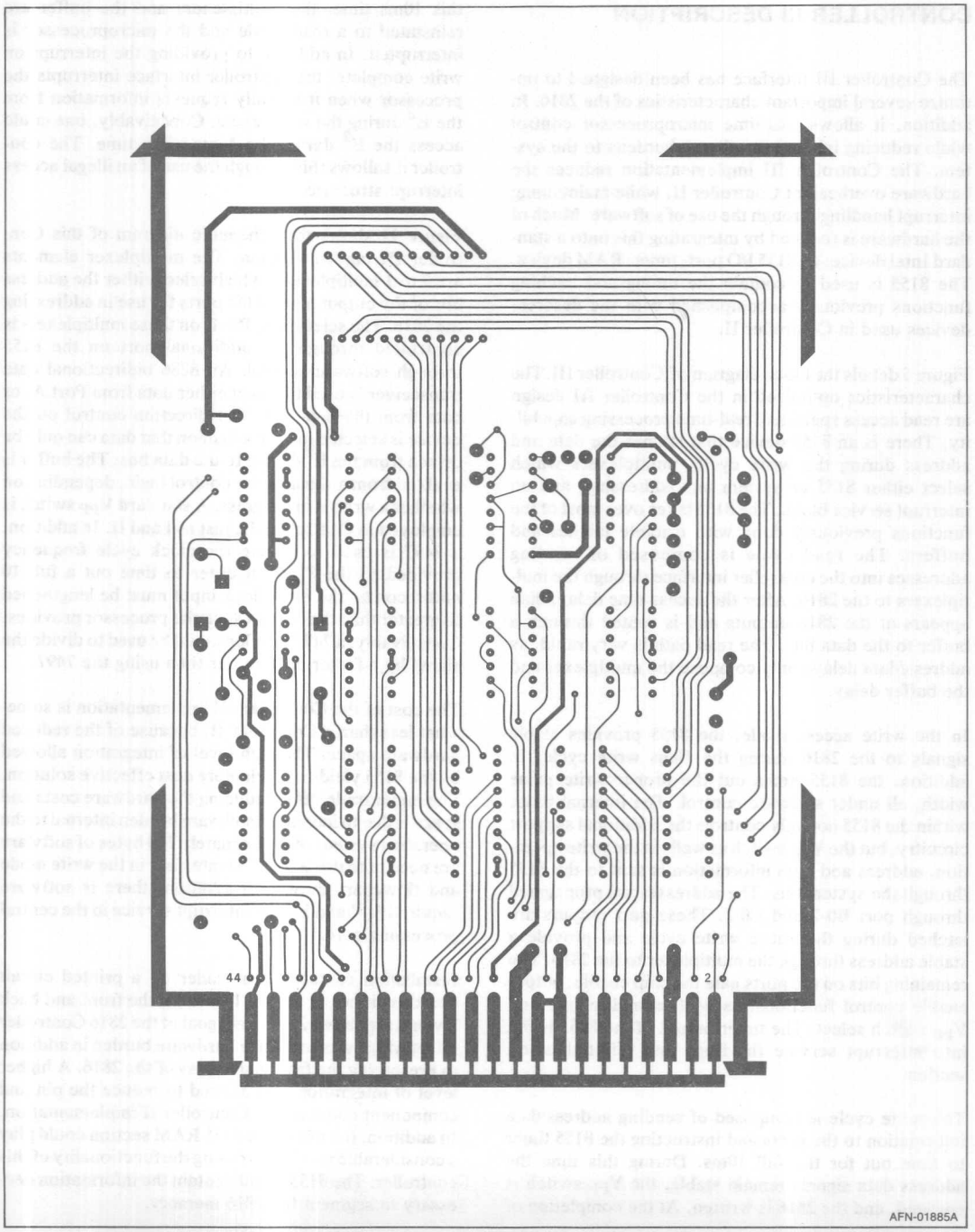



Figure 10c. $E^2$-Demo II Controller II (Continued)

## CONTROLLER III DESCRIPTION

The Controller III interface has been designed to optimize several important characteristics of the 2816. In addition, it allows real-time microprocessor control while reducing inherent hardware burdens to the system. The Controller III implementation reduces the hardware overhead of Controller II, while maintaining interrupt handling through the use of software. Much of the hardware is reduced by integrating this onto a standard Intel device; an 8155 I/O port, timer, RAM device. The 8155 is used to contain the timing and latching functions previously accomplished with the descrete devices used in Controller II.

Figure 5 details the block diagram of Controller III. The characteristics optimized in the Controller III design are read access speed and real-time processing capability. There is an 8155 device that latches the data and address during the write cycle, multiplexers which select either 8155 or system bus addressing, and an interrupt service block. The 8155 takes over most of the functions previously done with discrete latches and buffers. The read cycle is composed of sending addresses into the controller interface through the multiplexers to the 2816. After the access time delay, data appears at the 2816 outputs and is routed through a buffer to the data bus. The read path is very rapid, as address/data delays only compose the multiplexer and the buffer delay.

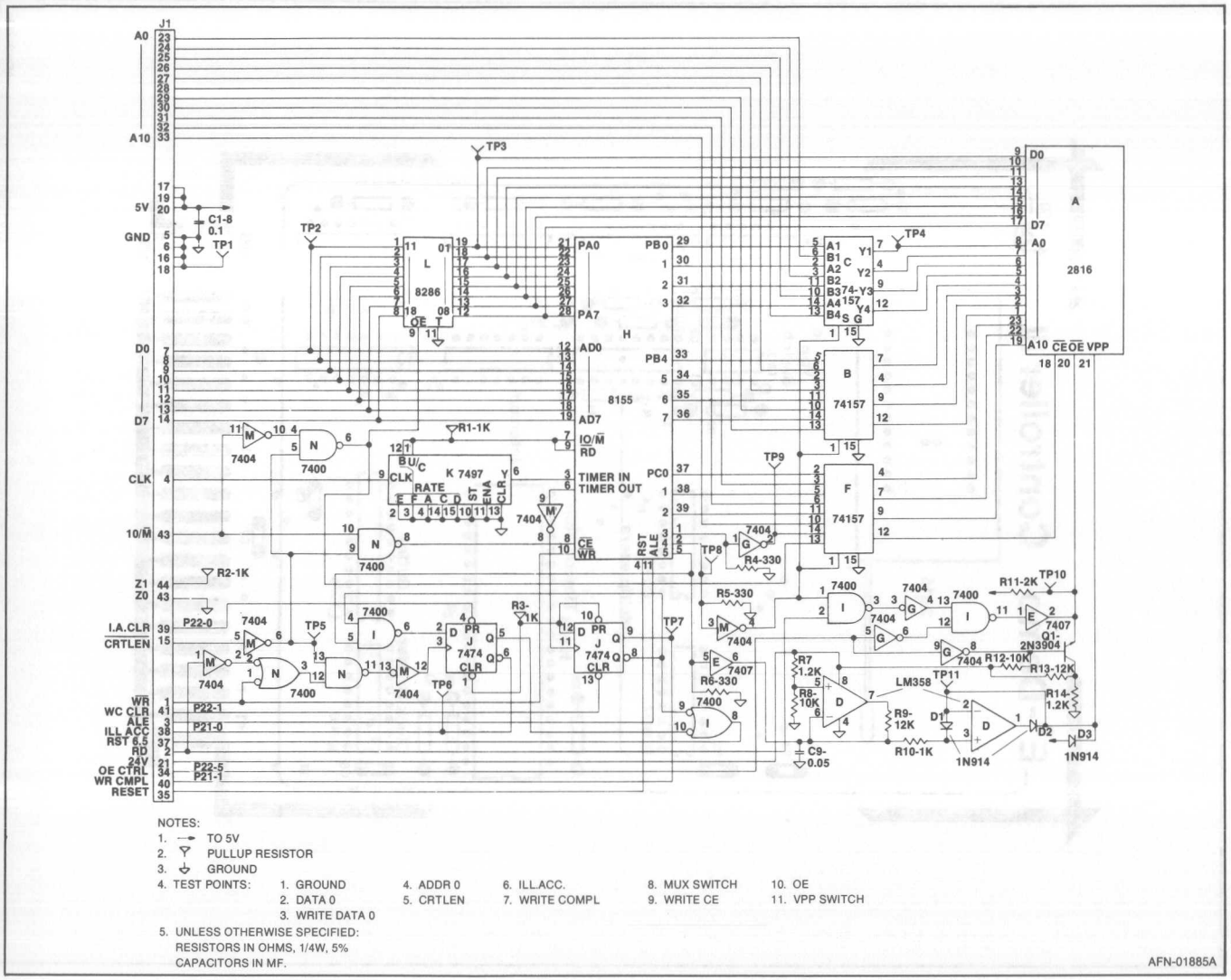
In the write access mode, the 8155 provides stable signals to the 2816 during the 10ms write cycle. In addition, the 8155 times out the proper write pulse width, all under software control. The internal timer within the 8155 not only controls the additional support circuitry, but the $V_{PP}$ switch as well. In the write operation, address and data information is sent to the 8155 through the system bus. The addresses are propagated through port B0-7 and C0-2. These port outputs are latched during the entire write cycle and provide a stable address through the multiplexer to the 2816. The remaining bits on the ports gate the chip enable, output enable control functions, as well as multiplexer and $V_{PP}$ switch select. The timer output of the 8155 is fed into interrupt service flip-flops and reinitialization section.

The write cycle is composed of sending address/data information to the ports and instructing the 8155 timer to time out for the full 10ms. During this time the address data signals remain stable, the $V_{PP}$ switch is engaged, and the 2816 is written. At the completion of this 10ms time, the multiplexers and the buffer are reinstated to a read mode and the microprocessor is interrupted. In addition to providing the interrupt on write complete, the controller interface interrupts the processor when it illegally requests information from the $E^2$ during the write cycle. Conceivably, one could access the $E^2$ during the 10ms write time. The controller disallows this through the use of an illegal access interrupt structure.

Figure 11 shows the schematic diagram of this Controller III implementation. The multiplexer elements are 2 to 1 multiplexers, which select either the address bus or the output of the 8155 ports for use in addressing the 2816. The select line, Pin 1, on these multiplexers is controlled through the additional port on the 8155 through software control. An 8286 bidirectional data transceiver is used to select either data from Port A, or data from the data bus. The direction control on the device is selected in such a fashion that data can only be driven from the $E^2$ device to the data bus. The buffer is enabled from a signal in the control logic, depending on whether a write is in progress. A standard $V_{PP}$ switch is employed in Controller III, just as I and II. In addition, a 7497 is used to reduce the clock cycle frequency provided to the 8155. In order to time out a full 10 milliseconds, the 8155 clock input must be lengthened to greater than the 320ns which the processor provides. Conceivably, a 7474 flip-flop could be used to divide the signal by a factor of 2, rather than using the 7497.

The cost of the Controller III implementation is somewhat less than a Controller II, because of the reduced hardware space. The high level of integration allowed by the 8155 yields a much more cost effective solution. The major trade-off in reducing the hardware costs and space is due to increased software burden internal to the operating system. Approximately 100 bytes of software are needed to drive the 8155 interface in the write mode and flowchart shown. In addition, there is software required for handling the interrupt service in the central processing core.

Installation of such a controller on a printed circuit board is shown in Figure 12, where the front and back layouts are shown. The main goal of the 2816 Controller III interface was to reduce hardware burden in addition to preserving the fast read access of the 2816. A higher level of integration was desired to reduce the pin and component count of the Controller II implementation. In addition, the use of the 8155 RAM section could play a considerable role in increasing the functionality of this controller. The 8155 could contain the information necessary to segment the 2816 memory.




NOTES:

1. TO 5V
2. PULLUP RESISTOR
3. GROUND
4. TEST POINTS:
   1. GROUND
   2. DATA 0
   3. WRITE DATA 0
   4. ADDR 0
   5. CRTLEN
   6. ILL.ACC.
   7. WRITE COMPL
   8. MUX SWITCH
   9. WRITE CE
   10. OE
   11. VPP SWITCH
5. UNLESS OTHERWISE SPECIFIED:
   RESISTORS IN OHMS, 1/4W, 5%
   CAPACITORS IN MF.

Figure 11. E²-Demo II Controller III

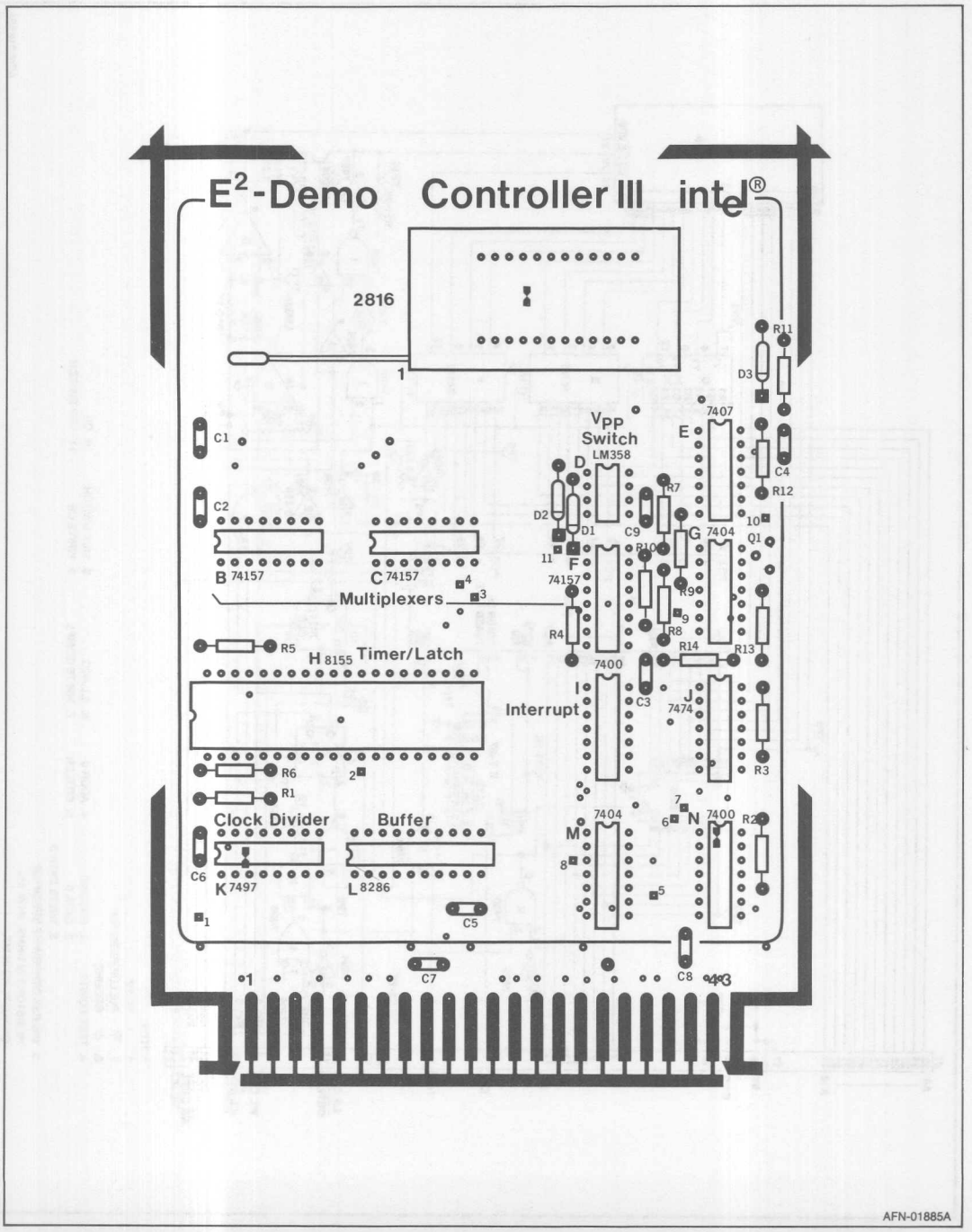
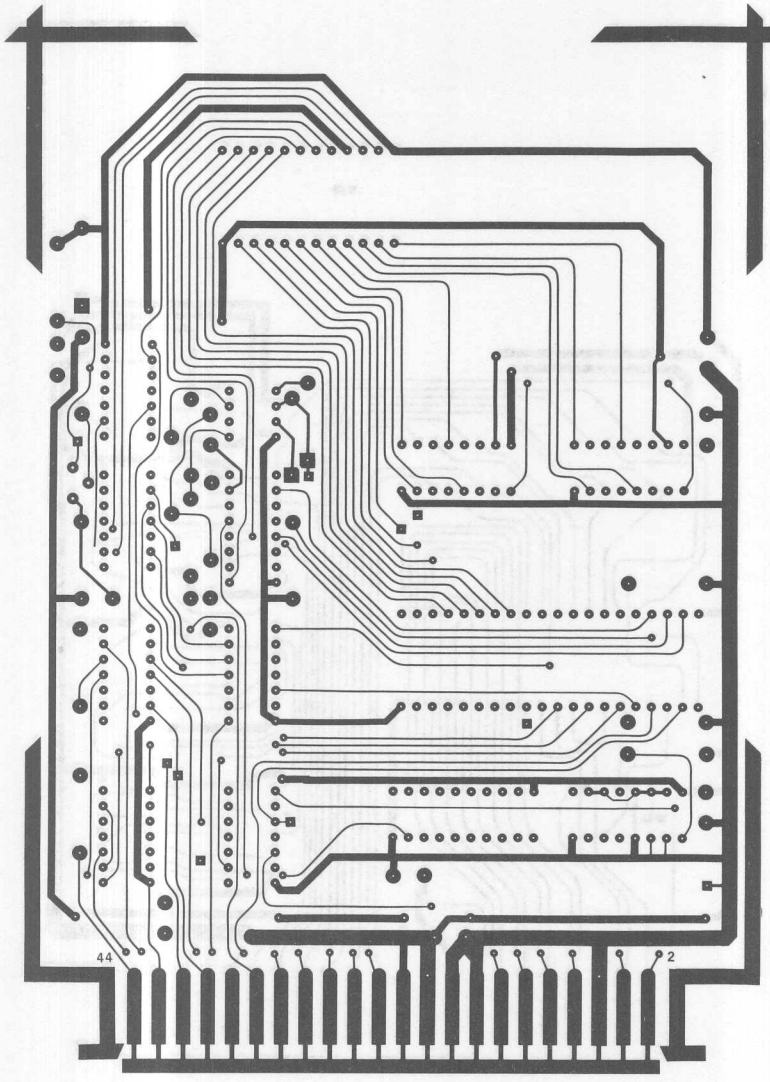



Figure 12a. E²-Demo Controller III



AFN-01913A




AFN-01885A

**Figure 12b. $E^2$-Demo Controller III (Continued)**



AFN-01913A

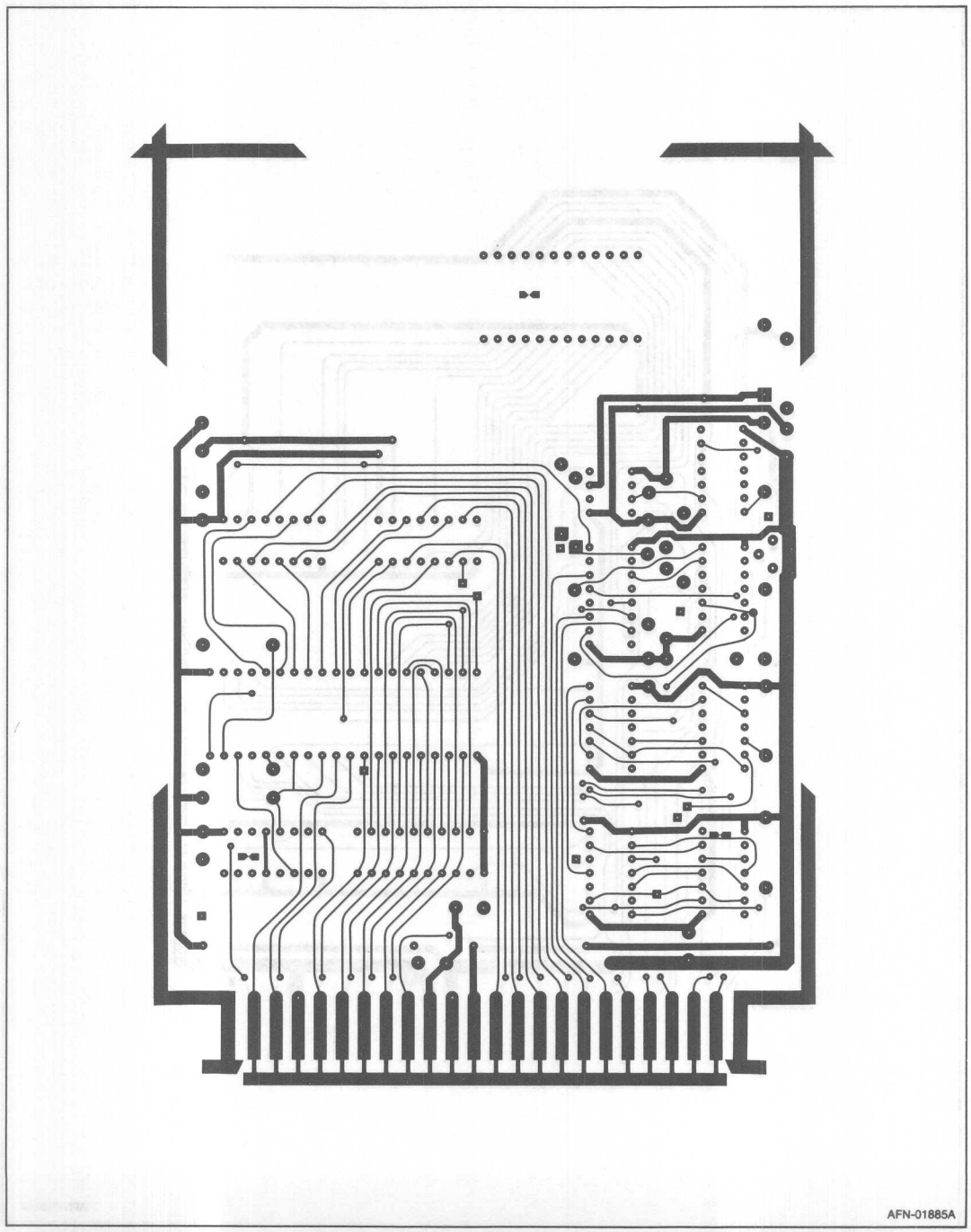



Figure 12c. $E^2$-Demo Controller III (Continued)

## CONTROLLER IV DESCRIPTION

Data collection is the key with the Controller IV implementation. The interface described here was designed to accommodate those designs in a data-logging or data-store application mode. The constraints for such a design are small size, relatively low power, and a high level of integration. Those constraints that are not of concern in a data-store application are read access time, where write time may be critical. An attempt was made to reduce the hardware burden associated with a data logging application, while maintaining a relatively efficient write access interaction. The read access time is the parameter that has been compromised for this design. In this case we use an I/O port, timer chip, as before, to cause latching of the signals for the 2816. However, the 8155 is utilized for both read and write operations. To read from the 2816, address/data information is sent into the 8155.

Addresses are sent into the 2816 through Port B and C, data is read back out from Port A. Since the I/O ports on the 8155 can be configured in either input or output modes, we can use one set for addresses and the other set for data. Data is brought back from the 2816 through the 8155 and placed on the multiplexed address/data bus. In order to write to the 2816 address, a software routine is set up which maps into the 8155 port.

Writing is accomplished by sending the address information through the address data bus into the 2816 through Ports B and C. The data is sent into Port A and is held latched while the write is in progress. Port C3 controls the chip enable function. Output enable and $V_{PP}$ drive are controlled by peripheral logic circuitry. To cause a write to the 2816, after the address/data information is loaded into the ports, the timer is commanded to time out. At the completion of the 10ms the processor is interrupted from the interrupt service block.

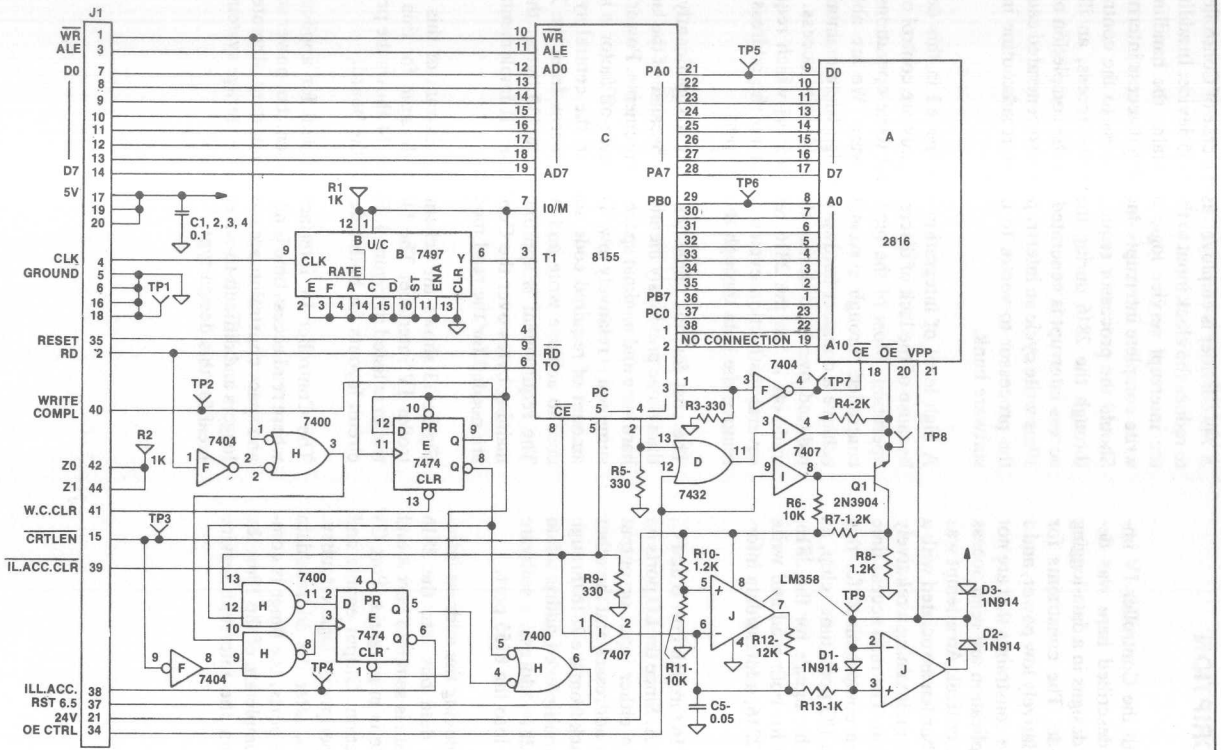
A 7497 divider is employed as the case of Controller III to reduce the clock input to the 8155 device. In addition, the interrupt service logic maintains the handling of write complete interrupts and illegal access interrupts. Should the processor request access to the controller through the 2816 during the write access, an illegal access interrupt is generated. At the completion of the 10ms write cycle an interrrupt is also generated causing the processor to vector to a restart subroutine in the software bank.

A high level of integration is achieved in this design because of the lack of discrete hardware control of the operation. Most of the read and write operations are controlled through system software. We are able to achieve a compact hardware design while maintaining reduced overhead during the 2816 write access. The trade-off is the the 2816 read access which requires several instruction cycles to set up the address and remove the data through the I/O port.

The cost for this implementation is significantly less than those previously mentioned because of the lack of hardware and minimal space requirements. Power consumption is relatively low. The trade-off factor is in the amount of required code space in the central system core to achieve write and read access from the 2816. The requirement is approximately 130 bytes, the remaining bytes over the Controller III implementation are needed for the read mode.

Figure 13 shows the schematic diagram of this Controller IV interface. The block diagram for this controller is listed in Figure 4. Figure 14 shows the printed circuit layouts for both sides of the board.

The Controller IV interface is ideal for applications where read access time is not critical, but power supply and space constraints are more important. Remote data loggers and difficult-to-access data storage systems are ideal for this design type.




NOTES:

1. → TO 5V
2. ▽ PULLUP RESISTOR
3. ⏚ GROUND
4. TEST POINTS:
   1. GROUND
   2. WRITE COMPLETE
   3. CRTLEN
   4. ILLEGAL ACCESS
   5. DATA 0
   6. ADDRESS 0
   7. CE
   8. OE
   9. VPP SWITCH
5. UNLESS OTHERWISE SPECIFIED:
   RESISTORS IN OHMS, 1/4W, 5%
   CAPACITORS IN MF.

AFN-01885A

**Figure 13. Controller IV Schematic Diagram**



AFN-01913A

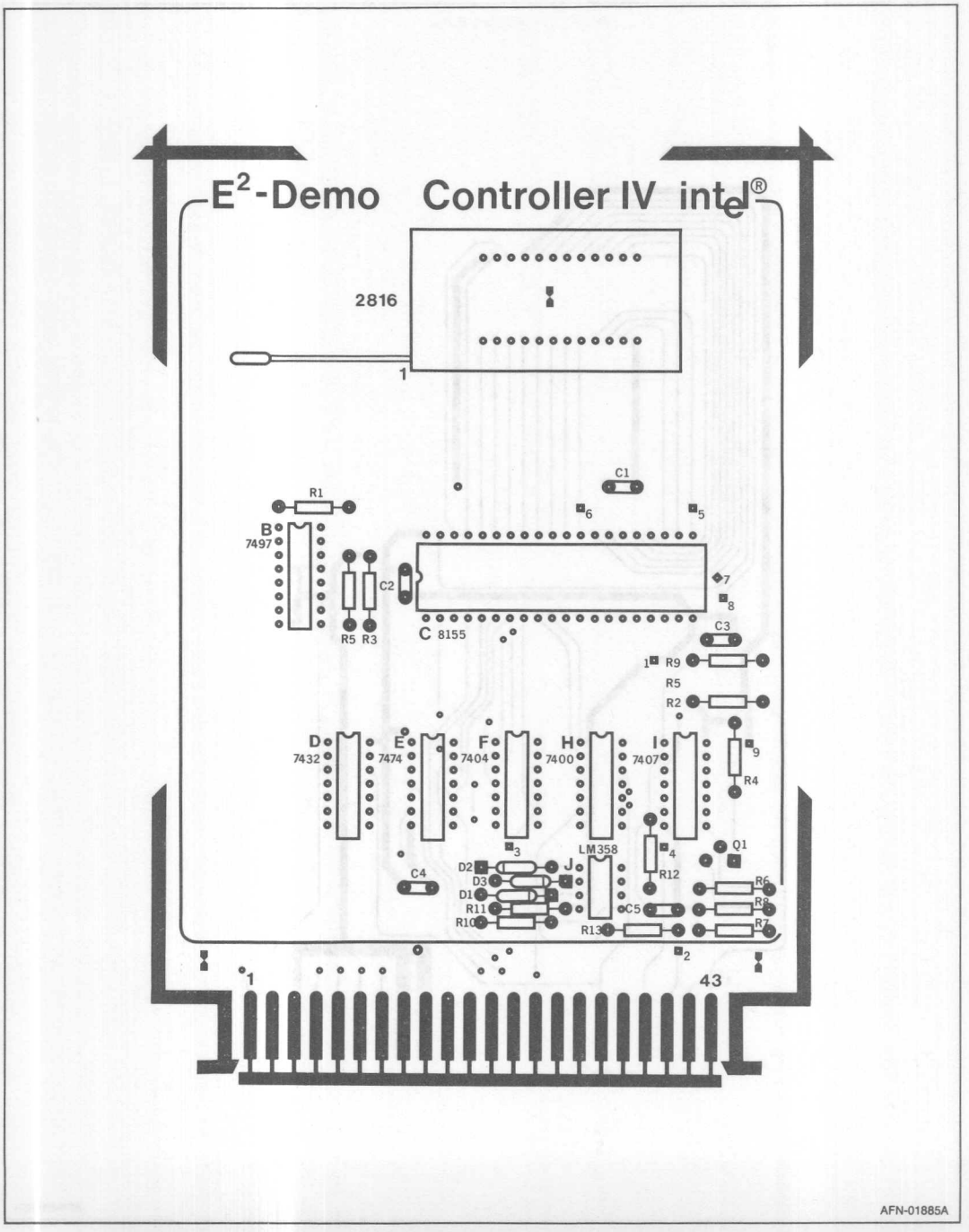



Figure 14a. E$^2$-Demo Controller IV

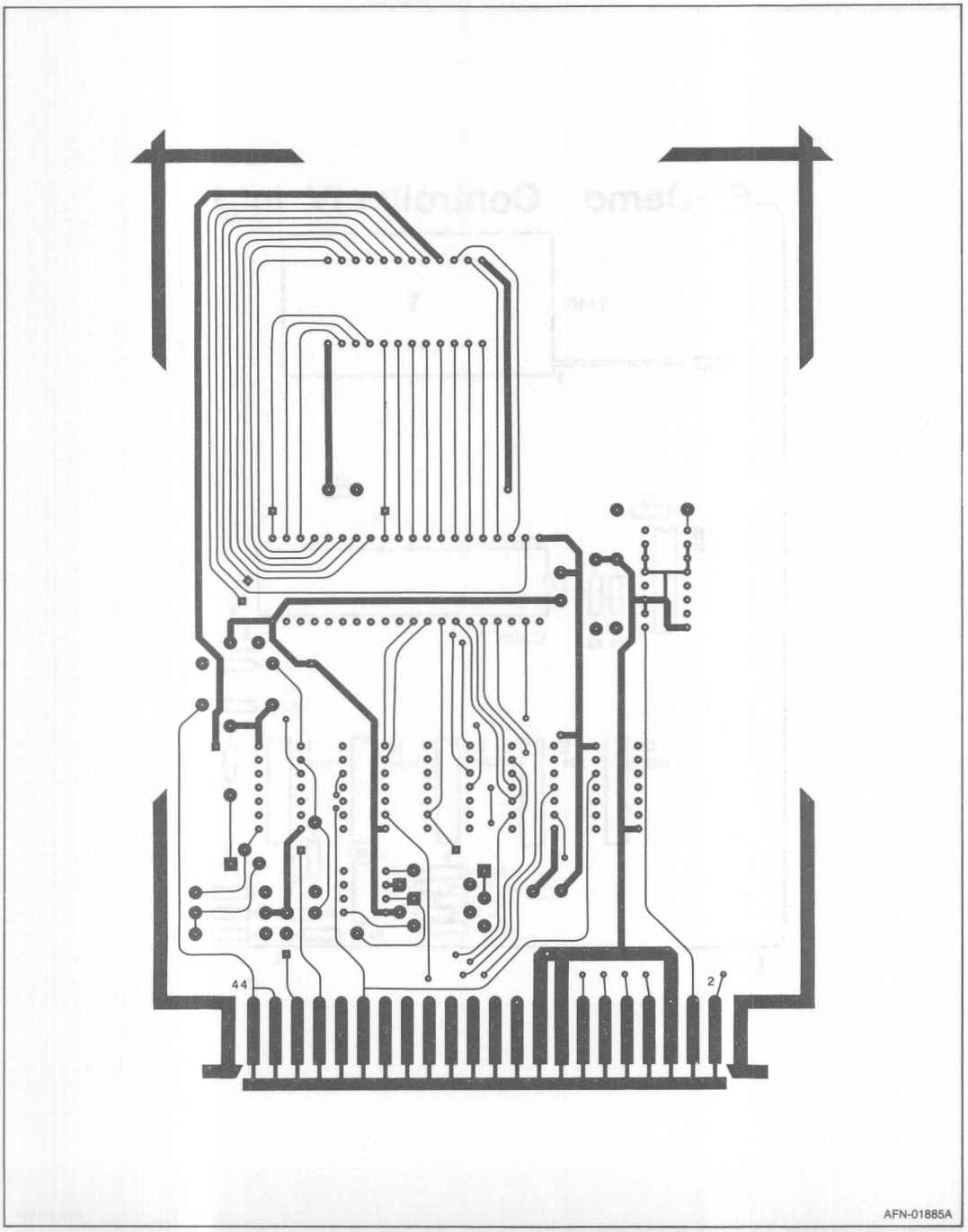



**Figure 14b. $E^2$-Demo Controller IV (Continued)**

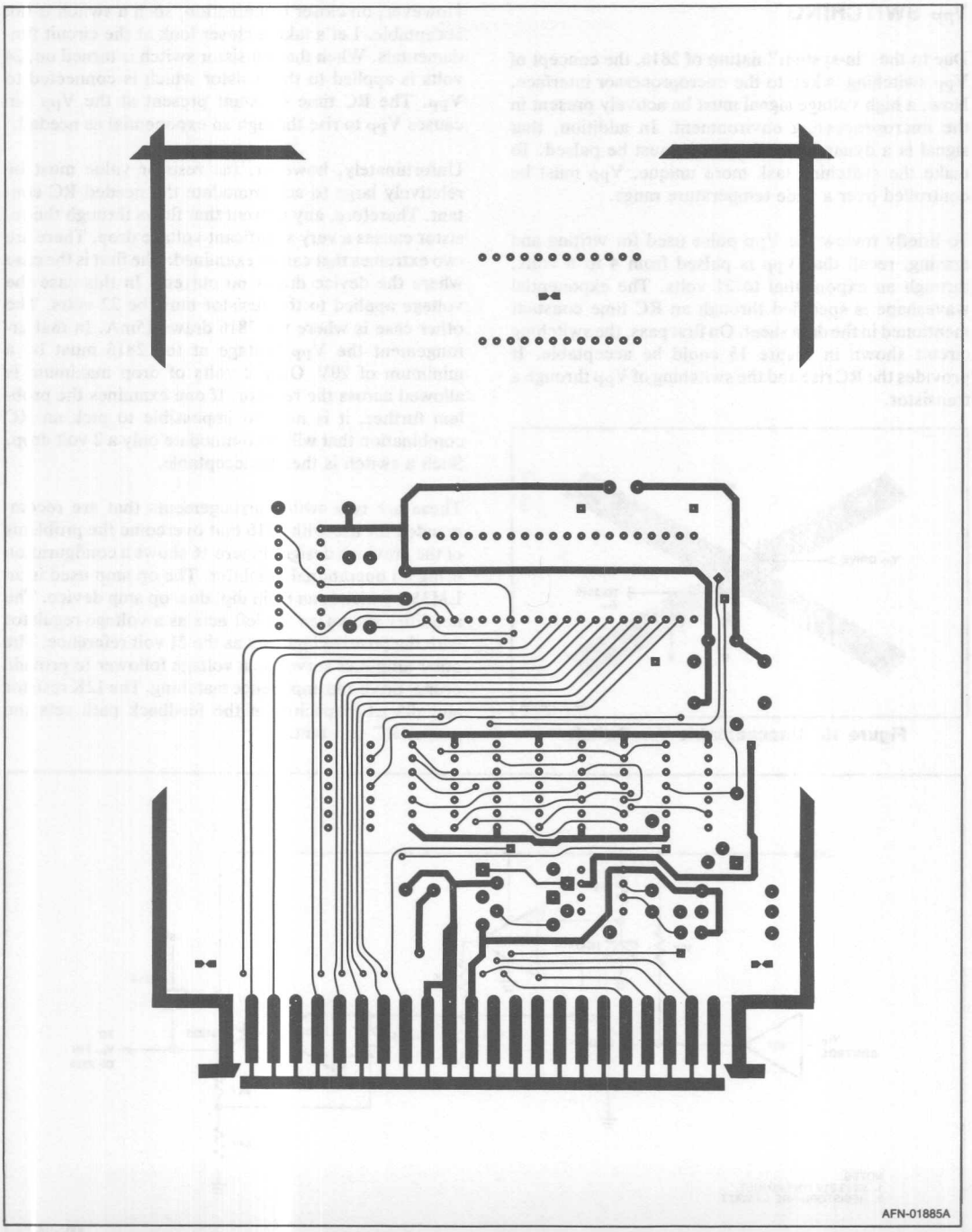



Figure 14c. $E^2$-Demo Controller IV (Continued)

## $V_{PP}$ SWITCHING

Due to the "in-system" nature of 2816, the concept of $V_{PP}$ switching is key to the microprocessor interface. Now, a high voltage signal must be actively present in the microprocessing environment. In addition, that signal is a dynamic one in that it must be pulsed. To make the switching task more unique, $V_{PP}$ must be controlled over a wide temperature range.

To briefly review the Vpp pulse used for writing and erasing, recall that $V_{PP}$ is pulsed from 4 to 6 volts, through an exponential to 21 volts. The exponential waveshape is specified through an RC time constant mentioned in the data sheet. On first pass, the switching circuit shown in Figure 15 could be acceptable. It provides the RC rise and the switching of $V_{PP}$ through a transistor.

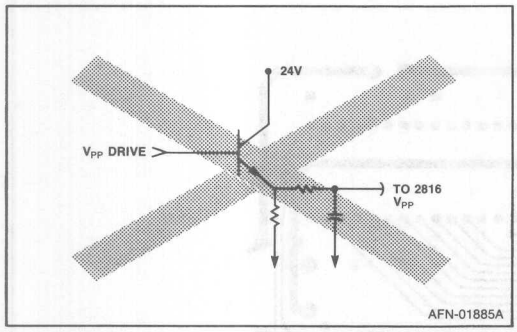



**Figure 15. Unacceptable $V_{PP}$ Switch**

However, on closer examination, such a switch is not acceptable. Let's take a closer look at the circuit fundamentals. When the transistor switch is turned on, 24 volts is applied to the resistor which is connected to $V_{PP}$. The RC time constant present at the $V_{PP}$ pin causes $V_{PP}$ to rise through an exponential as needed.

Unfortunately, however, the resistor value must be relatively large to accommodate the needed RC contant. Therefore, any current that flows through the resistor causes a very significant voltage drop. There are two extremes that can be examined: The first is the case where the device draws no current. In this case the voltage applied to the resistor must be 22 volts. The other case is where the 2816 draws 15mA. In that arrangement the $V_{PP}$ voltage at the 2816 must be a minimum of 20V. Only 2 volts of drop maximum is allowed across the resistor. If one examines the problem further, it is next to impossible to pick an RC combination that will accommodate only a 2 volt drop. Such a switch is then unacceptable.

These are two switch arrangements that are recommended for use with 2816 that overcome the problems of the previous design. Figure 16 shows a configuration using an operational amplifier. The op amp used is an LM358, which is an 8 pin dip, dual op amp device. The amplifier shown on the left acts as a voltage regulator with the positive input set as the 21 volt reference. The other amplifier serves as a voltage follower to provide proper drive and impedence matching. The 12K resistor and .05 $\mu$F capacitor in the feedback path sets the proper RC constant.

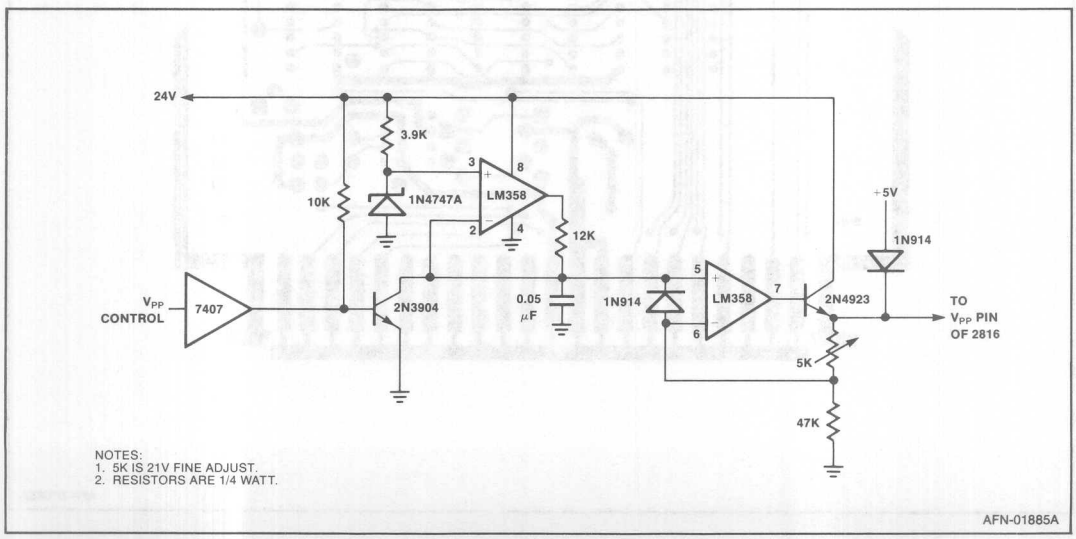



**Figure 16. Operational Amplifier Switch**

## $\overline{OE}$ SWITCHING

The 2816, in addition to byte erase functionality, can implement chip erase. All 2048 bytes can be erased in only 10ms. To accomplish this, however, requires application of a high voltage, ultra-low current signal to the $\overline{OE}$ pin. When the output enable pin is set into the range of 9-15 volts, and the $V_{PP}$ pin is pulsed to 21 volts, the entire chip is erased.

The current required at $\overline{OE}$ is a 10$\mu$A leakage, so little power is consumed. The switch shown in Figure 18 accomplishes switching $\overline{OE}$ to a higher voltage level when necessary. The chip erase control line can be derived from a port or other circuitry in the microprocessor system.

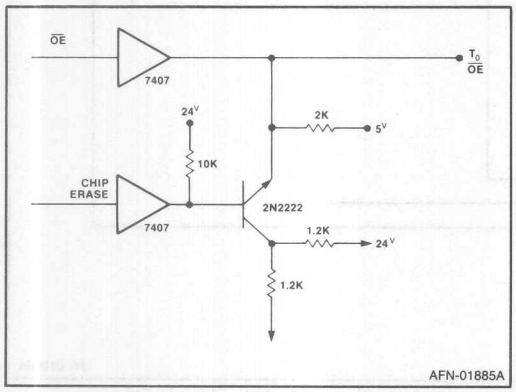



**Figure 18. Chip Erase Switch**

## MULTIPLE 2816's

Because of the flexibility of $E^2$, the capability to easily connect multiple devices together is essential. RAM's can be simply tied together, $E^2$ needs a similar functionality. Figure 19 shows the mode select for the 2816's write/erase inhibit mode.

| MODE \ PIN | $\overline{CE}$ (18) | $\overline{OE}$ (20) | $V_{PP}$ (21) | OUTPUTS |
|---|---|---|---|---|
| READ | $V_{IL}$ | $V_{IL}$ | +4 to +6 | $D_{OUT}$ |
| STANDBY | $V_{IH}$ | DON'T CARE | +4 to +6 | HIGH Z |
| BYTE ERASE | $V_{IL}$ | $V_{IH}$ | +21 | $D_{IN} = V_{IH}$ |
| BYTE WRITE | $V_{IL}$ | $V_{IH}$ | +21 | $D_{IN}$ |
| CHIP ERASE | $V_{IL}$ | +9 to +15V | +21 | $D_{IN} = V_{IH}$ |
| E/W INHIBIT | $V_{IH}$ | DON'T CARE | DON'T CARE | HIGH Z |

What this specification shows is that $V_{PP}$ can be at high voltage (21V) when the 2816 is deselected. From a system perspective, $V_{PP}$ can be bussed to multiple devices in the system. Any device that is to be written, can be, simply by TTL level control of $\overline{CE}$.

This allows simple and straightforward control of multiple 2816's in the system. Only one $V_{PP}$ switch is needed for the entire memory array, allowing a highly compact and cost effective design. Figure 20 shows how simple such an implementation can be.

## INTERFACE SOFTWARE REQUIREMENTS

As discussed, the various 2816 controllers employ various SSI and LSI devices. Each of the implementations require a varying degree of hardware and software. With Controller I, no software is necessary. Controller IV, on the other hand, needs approximately 130 bytes to handle the interface to the 8155 I/O port.

The following figures deal with the software drivers for the various controllers. These are several general subroutines that can be integrated in various ways, depending on the function and performance desired. Table 3 lists the various modules shown in the figures.

**Table 3.**

| | |
|---|---|
| Overall Write Subroutine | Figure 21 |
| Controller I Software Driver | Figure 22 |
| Controller II Software Driver | Figure 23 |
| Controller III Software Driver | Figure 24 |
| Controller IV Software Driver | Figure 25 |
| Controller II Chip Erase Routines | Figure 26 |
| Controller III, IV Chip Erase Routines | Figure 27 |
| Controller I/O Poll Routines | Figure 28 |
| Controller Interrupt Driver | Figure 29 |

Figure 21 shows the generalized write subroutine for all controllers. As indicated, data is passed through the 8085 A-register, and addresses passed through the HL-register pair. The routine first executes an erase, and then a write operation. The software driver that writes to the device is called WECYCL.

There is a unique WECYCL routine for each control interface. The driver for Controller I is a simple parameter pass routine, and a move to memory. This software is listed in Figure 22. The Controller II subroutine uses parameter pass, and interrupt initialization and service. The Controller II driver is listed in Figure 23. The interrupt service routine is given in Figure 29. In order to write to Controller III and IV interfaces, the 8155 I/O device must be initialized. A generalized flow chart for this operation is shown in Figure 24A. The software

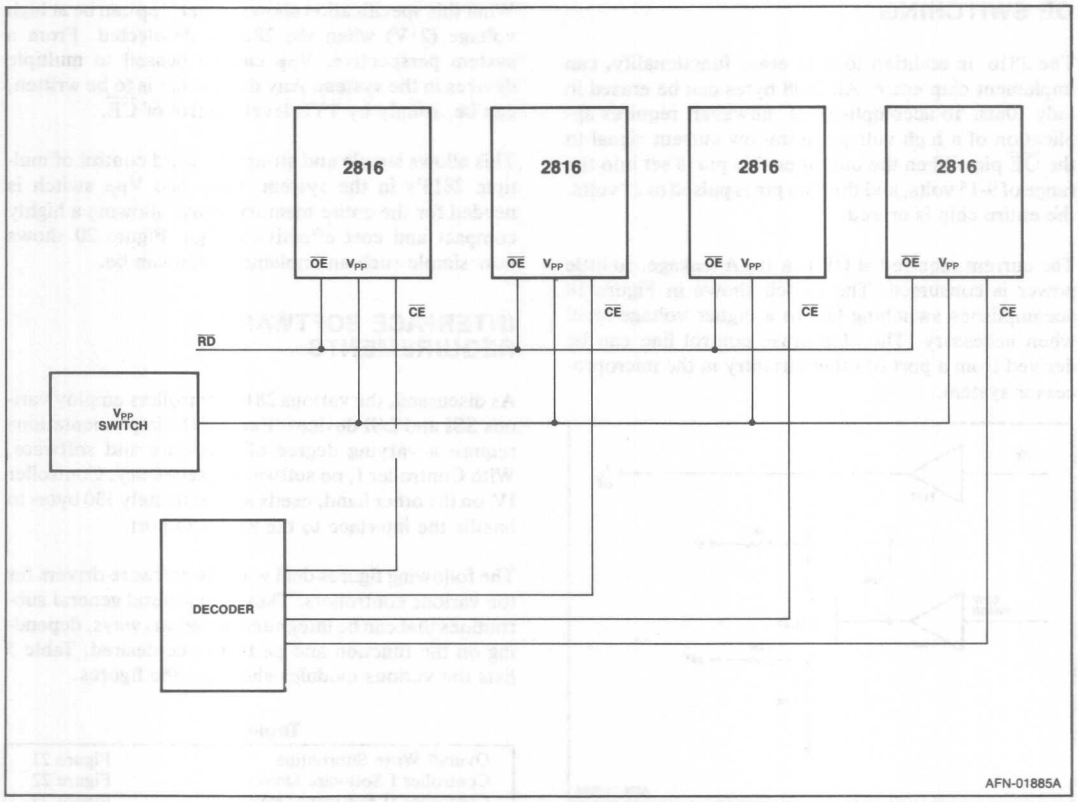



**Figure 20. Multiple 2816's In System**

listings are detailed in Figures 24 and 25. Both of these routines use the same interrupt service as Controller II. The remaining routines, for chip erase and I/O polling control, are shown in Figures 27, 28.

All of the interfaces, with the exception of the Controller IV, allow transparent reads of the 2816. Controller IV isolates the $E^2$ from the system bus through the 8155. A flowchart for Controller IV read operations is detailed in Figure 30.

## CONCLUSION

Based on the previous discussion, it is apparent that the interface to the 2816 is highly application dependent. Several interfaces have been presented, each of those optimized for a different system concern. Each of the controller implementations requires a different amount of hardware and software overhead, and provides a different throughput capability to the host processor. Each of these controllers is also appropriate for one or more design types. Controller I for program store areas, Controller IV for strict data store applications.

Controllers II and III are higher performance, and yet require a larger amount of hardware and software to service interrupts and generate 8155 timing controls. Further application notes will discuss some of the enhanced controllers, such as the bipolar state machine controller. All of these controllers are also available for test in the $E^2$ Demonstrator, which is a highly sophisticated tool for use with the 2816. The demonstrator is available by contacting a local Intel sales office and can be used for evaluation and test purposes of the $E^2$ device.

Above all, the interface to the CPU has been realized in a consistent and appropriate microprocessing architecture, something that has never been possible because of prior device attributes and technology constraints. The 2816 then adds an appropriate and applicable use of non-volatile and flexible memory to the current offerings of memory devices. It will prove a useful and powerful memory supplement and yield application and system benefits never before possible through consistent, convenient, and simple microprocessor interface.



AFN-01913A

```
ASM80 :F1:WRITE.SRC

ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ         LINE       SOURCE STATEMENT

                      1 $DEBUG
                      2
                      3
                      4
                      5         ;       ********************************************
                      6
                      7
                      8         ;               2816 CONTROLLER WRITE SUBROUTINE
                      9
                     10         ;
                     11         ;
                     12
                     13         ;       ********************************************
                     14
                     15
                     16         EXTRN   PECYCL
                     17
                     18         PUBLIC  WRITE
                     19
                     20         CSEG
                     21
                     22
                     23         ;       WRITE SUBROUTINE
                     24
                     25         ;       WRITES A BYTE TO THE 2816
                     26
                     27         ;       DATA PASSED:    A = DATA TO WRITE
                     28         ;                       HL = ADDRESS TO WRITE
                     29         ;       REGS DESTROYED: NONE
                     30         ;       CALLS:          PECYCL - PROGRAM/ERASE CYCLE SUBROUTINE
                     31
                     32 WRITE:
  0000 F5            33         PUSH    PSW             ; SAVE DATA WE'RE ABOUT TO WRITE
  0001 3EFF          34         MVI     A,0FFH          ; EXECUTE A BYTE ERASE FUNCTION
  0003 CD0000    E   35         CALL    PECYCL          ; BY WRITING 0FFH TO THE 2816
  0006 F1            36         POP     PSW             ; RESTORE DATA BYTE
  0007 CD0000    E   37         CALL    PECYCL          ; NOW WRITE THE DATA
  000A C9            38         RET                     ; AND RETURN
                     39
                     40         END

PUBLIC SYMBOLS
WRITE  C 0000

EXTERNAL SYMBOLS
PECYCL E 0000

USER S   JLS
PECYCL E 0000    WRITE  C 0000

ASSEMBLY COMPLETE,   NO ERRORS                                          AFN-01885A
```

**Figure 21. Overall Write Subroutine**



AFN-01913A

```
ASM80 :F1:CONT1.SRC MOD85

ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1


  LOC  OBJ        LINE       SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4        CSEG
                     5
                     6        PUBLIC  READ,WECYCL
                     7
                     8
                     9
                    10        ;       CONTROLLER I READ SUBROUTINE
                    11
                    12        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                    13        ;       DATA RETURNED:  A  = DATA READ
                    14        ;       REGS DESTROYED: NONE
                    15
                    16 READ:
  0000 7E           17        MOV     A,M             ; JUST READ FROM MEMORY
  0001 C9           18        RET
                    19
                    20
                    21
                    22
                    23        ;       CONTROLLER I WRITE/ERASE CYCLE SUBROUTINE
                    24
                    25        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                    26        ;                       A  = DATA TO WRITE
                    27        ;                            OR 0FFH  (ERASE)
                    28        ;       DATA RETURNED:  NONE
                    29        ;       REGS DESTROYED: NONE
                    30
                    31 WECYCL:
  0002 77           32        MOV     M,A             ; JUST WRITE TO MEMORY
  0003 C9           33        RET
                    34
                    35
                    36        END

PUBLIC SYMBOLS
READ   C 0000    WECYCL C 0002
EXTERNAL SYMBOLS

USER SYMBOLS
READ   C 0000    WECYCL C 0002
ASSEMBLY COMPLETE,   NO ERRORS
```

AFN-01885A

**Figure 22. Controller I Software Driver**



AFN-01913A

```
ASM80 :F1:CONT2.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ       LINE     SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4         PUBLIC  WECYCL,READ,ENDWE,CCLEAR
                    5
                    6         EXTRN   WEDELY
                    7
                    8         CSEG
                    9
                   10
                   11         ;       CONTROLLER II READ SUBROUTINE
                   12
                   13         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                   14         ;       DATA RETURNED:  A  = DATA READ
                   15         ;       REGS DESTROYED: NONE
                   16
                   17 READ:
  0000 7E          18         MOV     A,M             ; JUST READ FROM MEMORY
  0001 C9          19         RET
                   20
                   21
                   22
                   23
                   24         ;       CONTROLLER II WRITE/ERASE CYCLE SUBROUTINE
                   25
                   26         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                   27         ;                       A  = DATA TO WRITE
                   28         ;                                  OR 0FFH  (ERASE)
                   29         ;       DATA RETURNED:  NONE
                   30         ;       REGS DESTROYED: NONE
                   31         ;       CALLS:          WEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                   32
                   33         ;       I/O PORTS USED:
                   34         ;               PORT 22H (OUTPUT) - CONTAINS BITS USED FOR RESETTING
                   35         ;                                   CONTROLLER INTERRUPT FLIP/FLOPS
                   36         ;                       BIT 0 = WRITE COMPL RESET (ACTIVE LOW)
                   37         ;                       BIT 1 = ILL ACCESS  RESET (ACTIVE LOW)
                   38
                   39         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                   40         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                   41         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                   42         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                   43         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                   44         ;                       WECYCL.
                   45
                   46
                   47         ;       I/O SYMBOLS
                   48
  0022             49 CLRPRT  EQU     22H             ; I/O PORT USED TO CLEAR INTERRUPT F/F'S
  0000             50 CLRACT  EQU     0               ; BIT PATTERN TO ACTIVATE CLEAR FUNCTION
```

AFN-01885A

**Figure 23. Controller II Software Driver**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
 LOC  OBJ         LINE       SOURCE STATEMENT

 0003              51 CLRINA  EQU     3H             ; BIT PATTERN TO DE-ACTIVATE CLEAR FUNCTION
                   52
                   53 WECYCL:
 0002 F5           54         PUSH    PSW            ; SAVE DATA TO WRITE
 0003 3E00         55         MVI     A,CLRACT       ; CLEAR WRITE COMPLETE AND ILLEGAL ACCESS F/F'S
 0005 D322         56         OUT     CLRPRT
 0007 3E03         57         MVI     A,CLRINA       ; DE-ACTIVATE CLEAR FUNCTION
 0009 D322         58         OUT     CLRPRT
 000B F1           59         POP     PSW            ; RESETORE DATA TO WRITE
 000C 77           60         MOV     M,A            ; JUST WRITE TO MEMORY
 000D CD0000   E   61         CALL    WEDELY         ; GO TO I/O POLL ROUTINE OR INTERRUPT DRIVER
 0010 C9           62         RET
                   63
                   64
                   65
                   66
                   67         ;       CONTROLLER II CHIP CLEAR SUBROUTINE
                   68
                   69         ;       DATA PASSED:    NONE
                   70         ;       DATA RETURNED:  NONE
                   71         ;       REGS DESTROYED: NONE
                   72         ;       CALLS           PEDELY ( I/O POLL ROUTINE OR INTERRUPT DRIVER)
                   73
                   74         ;       I/O PORTS USED:
                   75         ;                       PORT 22H (OUTPUT)
                   76         ;                               BIT 0 = WRITE COMPLETE CLEAR (ACTIVE LOW)
                   77         ;                               BIT 1 = ILLEGAL ACCESS CLEAR (ACTIVE LOW)
                   78         ;                               BIT 5 = CHIP CLR (+12V TO OE' LINE) (ACTIVE HI)
                   79
                   80         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                   81         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                   82         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                   83         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                   84         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                   85         ;                       CCLEAR.
                   86
                   87
                   88         ;       I/O SYMBOLS
                   89
 0023              90 CLRCCL  EQU     23H            ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                   91                                ; OE' = +12V FUNCTION FOR CHIP CLEAR
                   92 CCLEAR:
 0011 F5           93         PUSH    PSW            ; SAVE REGISTERS
 0012 E5           94         PUSH    H
 0013 3E00         95         MVI     A,CLRACT       ; GET BITS TO RESET WRITE COMPL AND ILL ACC
 0015 D322         96         OUT     CLRPRT
 0017 3E23         97         MVI     A,CLRCCL       ; GET BITS TO DEACTIVATE CLEAR FUNCTION AND
                   98                                ; TURN ON OE' = +12V FUNCTION FOR CHIP CLEAR
 0019 D322         99         OUT     CLRPRT         ; OUTPUT TO I/O PORT
 001B 3EFF        100         MVI     A,0FFH         ; WRITE 0FFH TO THE 2816
 001D 3200A0      101         STA     0A000H                                          AFN-01885A
```

**Figure 23. Controller II Software Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
 LOC  OBJ         LINE       SOURCE STATEMENT

 0020 CD0000   E   102         CALL    WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
 0023 3E03         103         MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
 0025 D322         104         OUT     CLRPRT
 0027 E1           105         POP     H               ; RESTORE REGISTERS
 0028 F1           106         POP     PSW
 0029 C9           107         RET
                   108
                   109
                   110
                   111         ;       CONTROLLER II END-OF-WRITE/ERASE-CYCLE ROUTINE
                   112
                   113         ;       JUMPED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                   114         ;       TO SHUT DOWN CONTROLLER.
                   115
                   116 ENDWE:
 002A C9           117         RET             ; JUST RETURN NORMALLY - NOTHING TO SHUT DOWN.
                   118
                   119         END

PUBLIC SYMBOLS
CCLEAR C 0011    ENDWE  C 002A    READ   C 0000    WECYCL C 0002

EXTERNAL SYMBOLS
WEDELY E 0000
USER SYMBOLS
CCLEAR C 0011    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    ENDWE  C 002A    READ   C 0000
WECYCL C 0002    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS
```

AFN-01885A

**Figure 23. Controller II Software Driver (Continued)**



AFN-01913A

```
ASM80 :F1:CONT3.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE      SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4         CSEG
                    5
                    6
                    7         PUBLIC  WECYCL,READ,ENDWE
                    8
                    9         EXTRN   WEDELY
                   10
                   11
                   12         ;       CONTROLLER III I/O PORT DEFINITIONS
                   13
                   14         ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                   15
                   16
                   17         ;       PORT    DESCRIPTION
                   18         ;       ----    -----------
                   19         ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                   20         ;
                   21         ;       0A1H    2816 DATA (OUTPUT)
                   22         ;
                   23         ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                   24         ;
                   25         ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                   26         ;                       BITS 0-2:       A8-A10
                   27         ;                       BIT  3:         CE CTRL (0=SELECT READ,
                   28         ;                                                WRITE ENABLE)
                   29         ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                   30         ;                       BIT 5:          VPP CTRL (0=INACTIVE,1=ACTIVE)
                   31         ;
                   32         ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                   33         ;
                   34         ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                   35         ;
                   36         ;       22H     CLEAR INTERRUPT FLIP-FLOPS PORT (OUTPUT)
                   37         ;                       BIT 0:          WRITE COMPL CLEAR (ACTIVE LOW)
                   38         ;                       BIT 1:          ILLEGAL ACC CLEAR (ACTIVE LOW)
                   39         ;                       BIT 5:          CHIP CLEAR MODE   (ACTIVE HI)
                   40
                   41
 00A0              42 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
 00A1              43 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
 00A2              44 ADRPRT  EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
 00A3              45 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
 00A4              46 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
 00A5              47 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                   48
 00C0              49 COUNTL  EQU     0C0H            ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
 0083              50 COUNTH  EQU     83H             ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
```

AFN-01885A

**Figure 24. Controller III Software Driver**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

                   51
                   52
                   53         ;       CONTROLLER III READ SUBROUTINE
                   54
                   55         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                   56         ;       DATA RETURNED:  A  = DATA READ
                   57         ;       REGS DESTROYED: NONE
                   58
                   59 READ:
 0000 7E           60         MOV     A,M             ; JUST READ FROM MEMORY
 0001 C9           61         RET     ;
                   62
                   63
                   64
                   65
                   66         ;       CONTROLLER III WRITE/ERASE CYCLE SUBROUTINE
                   67
                   68         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                   69         ;                       A  = DATA TO WRITE
                   70         ;                                  OR 0FFH  (ERASE)
                   71         ;       DATA RETURNED:  NONE
                   72         ;       REGS DESTROYED: NONE
                   73         ;       RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                   74         ;       CALLS:          PEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                   75         ;
                   76         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                   77         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                   78         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                   79         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                   80         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                   81         ;                       PECYCL.
                   82
                   83
 0000              84 CLRACT  EQU     0H              ; ACTIVE CLEAR WRITE COMPL & ILL ACC FUNCTION
 0003              85 CLRINA  EQU     3H              ; INACTIVE CLEAR WC & IA FUNCTION
 0022              86 CLRPRT  EQU     22H             ; PORT USED TO CLEAR ILL ACC & WRT COMPL F/F
                   87
                   88
                   89 WECYCL:
 0002 F5           90         PUSH    PSW             ; SAVE REGISTERS
 0003 C5           91         PUSH    B
 0004 47           92         MOV     B,A             ; SAVE DATA TO WRITE IN B-REGISTER
 0005 3E00         93         MVI     A,CLRACT        ; CLEAR WRITE COMPLETE AND ILL ACC FLIP-FLOPS
 0007 D322         94         OUT     CLRPRT
 0009 3E03         95         MVI     A,CLRINA        ; DE-ACTIVATE CLEAR FUNCTION
 000B D322         96         OUT     CLRPRT
 000D 3E0F         97         MVI     A,0FH           ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 000F D3A0         98         OUT     EEPDR           ; OUTPUT TO PORT DIRECTION REGISTER
 0011 78           99         MOV     A,B             ; FETCH DATA TO WRITE
 0012 D3A1        100         OUT     DATPRT          ; OUTPUT TO 2816 DATA LINES
 0014 7D          101         MOV     A,L             ; GET LOW ORDER ADDRES                    AFN-01885A
```

**Figure 24. Controller III Software Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
 LOC  OBJ          LINE        SOURCE STATEMENT

 0015 D3A2          102         OUT     ADRPRT          ; OUTPUT TO ADDRESS LINES
 0017 7C            103         MOV     A,H             ; GET HIGH ORDER ADDRESS
 0018 E607          104         ANI     7H              ; CLEAR ALL CONTROL LINES
 001A F610          105         ORI     10H             ; ADD MUX BIT TO SELECT I/O PORTS FOR WRITE
 001C D3A3          106         OUT     CTLPRT          ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 001E F608          107         ORI     8H              ; ADD CE ACTIVE BIT
 0020 D3A3          108         OUT     CTLPRT          ; OUTPUT CONTROL LINES AGAIN
 0022 47            109         MOV     B,A             ; SAVE HIGH ORDER ADDR/CTL LINE DATA
 0023 3EC0          110         MVI     A,COUNT         ; OUTPUT TIMER COUNT (LOW ORDER)
 0025 D3A4          111         OUT     TIMLOW
 0027 3E83          112         MVI     A,COUNTH        ; OUTPUT TIMER COUNT (HIGH ORDER)
 0029 D3A5          113         OUT     TIMHI
 002B 3ECF          114         MVI     A,0CFH          ; START THE TIMER
 002D D3A0          115         OUT     EEPDR
 002F 78            116         MOV     A,B             ; RETRIEVE ADDRESS/CONTROL BITS
 0030 F620          117         ORI     20H             ; ADD VPP ACTIVE BIT
 0032 D3A3          118         OUT     CTLPRT          ; ACTIVATE VPP
 0034 320000    D   119         STA     TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
 0037 CD0000    E   120         CALL    WEDELY          ; WAIT FOR END OF WRITE CYCLE BY I/O POLL OR
                    121                                 ; INTERRUPT DRIVER ROUTINE
 003A C1            122         POP     B               ; RESTORE REGISTERS
 003B F1            123         POP     PSW
 003C C9            124         RET                     ; BACK TO CALLING ROUTINE
                    125
                    126
                    127
                    128         DSEG
                    129
 0000               130 TEMCTL: DS      1               ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                    131
                    132
                    133
                    134         CSEG
                    135
                    136
                    137
                    138         ;       CONTROLLER III END-OF-WRITE/ERASE-CYCLE ROUTINE
                    139
                    140         ;       JUMPED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                    141         ;       TO SHUT DOWN CONTROLLER.
                    142
                    143         ;       DATA PASSED:    TEMCTL (1 RAM BYTE) CONTAINING HIGH ORDER
                    144         ;                       ADDRESS (3 BITS) & CONTROL BEFORE WRITE COMPL.
                    145
                    146 ENDWE:
 003D F5            147         PUSH    PSW             ; SAVE REGISTERS WE'LL DESTROY
 003E D5            148         PUSH    D
 003F 3A0000    D   149         LDA     TEMCTL          ; GET ADDRESS LINES/CONTROL BITS
 0042 E61F          150         ANI     1FH             ; REMOVE ACTIVE VPP BIT
 0044 D3A3          151         OUT     CTLPRT          ; DE-ACTIVATE VPP
                    152                                                          AFN-01885A
```

**Figure 24. Controller III Software Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   4
  LOC  OBJ        LINE       SOURCE STATEMENT

  0046 F5          153        PUSH    PSW            ; SAVE HIGH ORDER ADDRESS/CONTROL LINES
  0047 110D00      154        LXI     D,13D          ; SET UP COUNT FOR 100 USEC DELAY
                   155 DELAY:
  004A 1B          156        DCX     D              ; DELAY WHILE VPP FALLS
  004B 7A          157        MOV     A,D            ; DONE COUNTING?
  004C B3          158        ORA     E
  004D C24A00   C  159        JNZ     DELAY          ; NO: KEEP LOOPING
                   160
  0050 F1          161        POP     PSW            ; RESTORE ADDRESS/CONTROL LINES
  0051 E617        162        ANI     17H            ; REMOVE CE ACTIVE BIT
  0053 D3A3        163        OUT     CTLPRT         ; DE-ACTIVATE CE
  0055 E607        164        ANI     7H             ; REMOVE MUX SELECT WRITE BIT
  0057 D3A3        165        OUT     CTLPRT         ; LET MUX SELECT FOR READ OPERATIONS
  0059 3E0E        166        MVI     A,0EH          ; PUT DATA PORT BACK TO INPUT MODE
  005B D3A0        167        OUT     EEPDR          ; SO AS NOT TO CAUSE CONTENTION W/ DATA BUS
                   168
  005D D1          169        POP     D              ; RESTORE REGISTERS
  005E F1          170        POP     PSW
  005F C9          171        RET                    ; AND EXIT
                   172
                   173
                   174
                   175        END


PUBLIC SYMBOLS
ENDWE  C 003D    READ   C 0000    WECYCL C 0002

EXTERNAL SYMBOLS
WEDELY E 0000
USER SYMBOLS
ADRPRT A 00A2    CLRACT A 0000    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083    COUNTL A 00C0    CTLPRT A 00A3
DATPRT A 00A1    DELAY  C 004A    EEPDR  A 00A0    ENDWE  C 003D    READ   C 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WECYCL C 0002    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS                                                          AFN-01885A
```

**Figure 24. Controller III Software Driver (Continued)**



AFN-01913A

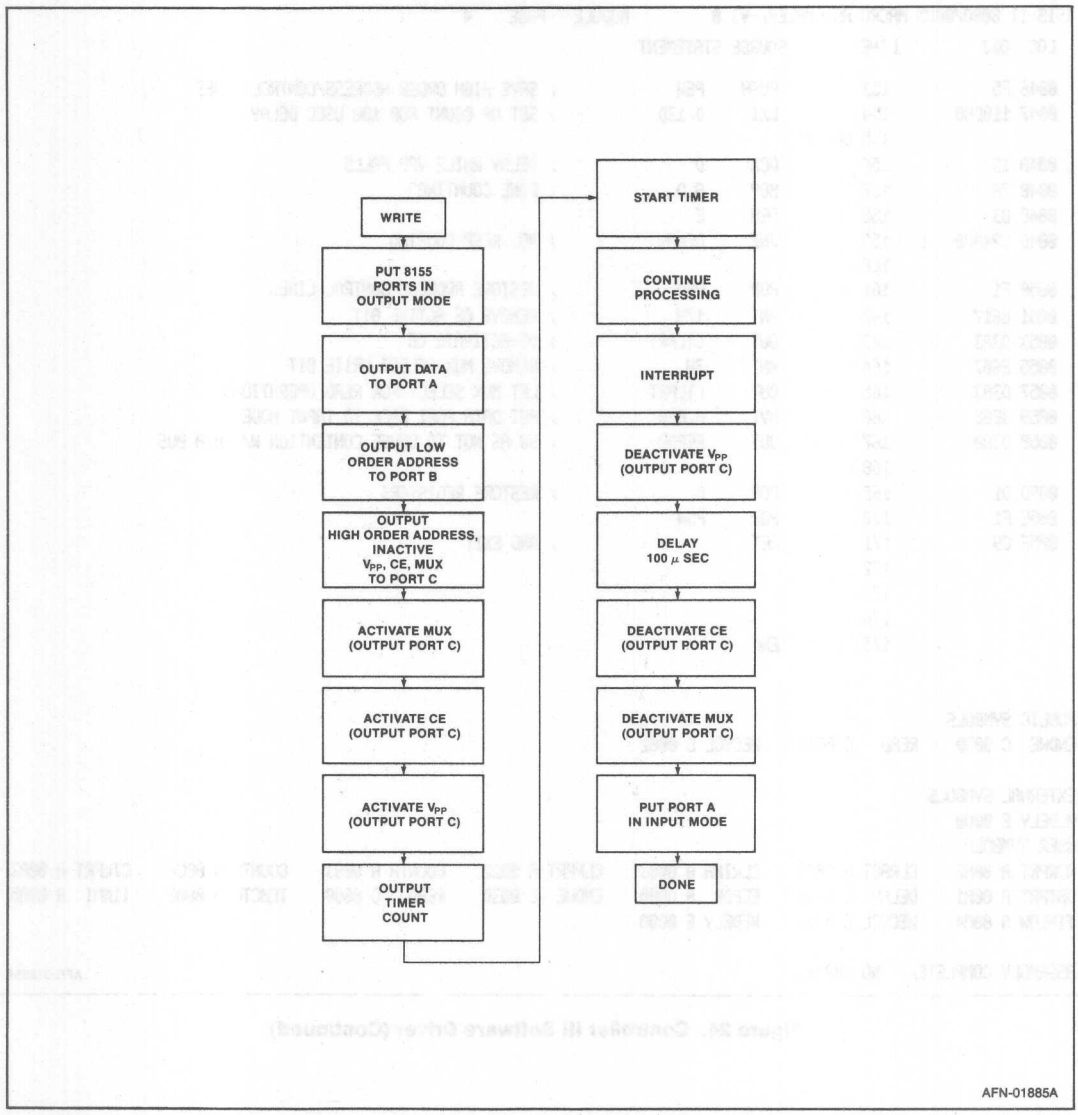



**Figure 24A. Controller III, IV, Flowchart**



AFN-01913A

```
ASM80 :F1:CONT4.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ        LINE        SOURCE STATEMENT
                     1 $DEBUG
                     2
                     3
                     4         PUBLIC  WECYCL,READ,ENDWE
                     5
                     6         EXTRN   WEDELY
                     7
                     8         CSEG
                     9
                    10
                    11         ;       CONTROLLER IV I/O PORT DEFINITIONS
                    12
                    13         ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                    14
                    15
                    16         ;       PORT    DESCRIPTION
                    17         ;       ----    -----------
                    18         ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                    19         ;
                    20         ;       0A1H    2816 DATA (OUTPUT)
                    21         ;
                    22         ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                    23         ;
                    24         ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                    25         ;                       BITS 0-2:       A8-A10
                    26         ;                       BIT  3:         CE CTRL (0=SELECT READ,
                    27         ;                                               WRITE ENABLE)
                    28         ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                    29         ;                       BIT 5:          VPP CTRL (0=INACTIVE,1=ACTIVE)
                    30         ;
                    31         ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                    32         ;
                    33         ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                    34         ;
                    35         ;       22H     PORT USED TO CLEAR WRITE COMPL & ILLEGAL ACC INTERRUPTS
                    36
                    37
  00A0              38 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
  00A1              39 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
  00A2              40 ADRPRT  EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
  00A3              41 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
  00A4              42 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
  00A5              43 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                    44
  00C0              45 COUNTL  EQU     0C0H            ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
  0083              46 COUNTH  EQU     83H             ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
                    47
                    48
                    49         ;       CONTROLLER IV READ SUBROUTINE
                    50
```

AFN-01885A

**Figure 25. Controller IV Software Driver**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
  LOC  OBJ         LINE        SOURCE STATEMENT

                     51        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO READ
                     52        ;       DATA RETURNED:  A  = DATA READ
                     53        ;       REGS DESTROYED: FLAGS
                     54
                     55 READ:
  0000 3E0E          56        MVI     A,0EH           ; PUT DATA PRT IN INPUT MODE, ALL OTHERS-OUTPUT
  0002 D3A0          57        OUT     EEPDR           ; OUTPUT TO PORT DIRECTRION REGISTER
  0004 7D            58        MOV     A,L             ; GET LOW ORDER ADDRESS
  0005 D3A2          59        OUT     ADRPRT          ; OUTPUT TO ADDRESS PORT
  0007 7C            60        MOV     A,H             ; GET HIGH ORDER ADDRESS
  0008 E607          61        ANI     7H              ; REMOVE ALL CONTROL BITS (KEEP 3 BIT ADDRESS)
  000A F610          62        ORI     10H             ; ADD OE' INACTIVE BIT
  000C D3A3          63        OUT     CTLPRT          ; OUTPUT TO CONTROL PORT
  000E E607          64        ANI     07H             ; REMOVE OE' INACTIVE BIT (ACTIVATE OE)
  0010 F608          65        ORI     8H              ; ADD CE' ACTIVE BIT
  0012 D3A3          66        OUT     CTLPRT          ; OUTPUT TO CONTROL PORT
  0014 DBA1          67        IN      DATPRT          ; INPUT DATA FROM 2816
  0016 F5            68        PUSH    PSW             ; SAVE DATA
  0017 AF            69        XRA     A               ; ZERO A REGISTER
  0018 D3A3          70        OUT     CTLPRT          ; DEACTIVATE ALL CONTROL LINES
  001A F1            71        POP     PSW             ; RESTORE DATA
  001B C9            72        RET                     ; AND EXIT
                     73
                     74
                     75
                     76
                     77        ;       CONTROLLER IV WRITE/ERASE CYCLE SUBROUTINE
                     78
                     79        ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                     80        ;                       A  = DATA TO WRITE
                     81        ;                            OR 0FFH  (ERASE)
                     82        ;       DATA RETURNED:  NONE
                     83        ;       REGS DESTROYED: NONE
                     84        ;       RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                     85        ;       CALLS:          WEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                     86        ;
                     87        ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                     88        ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                     89        ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                     90        ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                     91        ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                     92        ;                       WECYCL.
                     93
                     94
  0022               95 CLRPRT  EQU     22H             ; I/O PORT USED TO RESET INTERRUPT F/F'S
  0000               96 CLRACT  EQU     0H              ; ACTIVATE CLEAR WRITE COMPL & ILL ACC INTR
  0003               97 CLRINA  EQU     3H              ; INACTIVE CLEAR WC & IA FUNCTION
                     98
                     99
                    100 WECYCL.
  001C F5           101        PUSH    PSW             ; SAVE REGISTERS                    AFN-01885A
```

**Figure 25. Controller IV Software Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
 LOC  OBJ          LINE        SOURCE STATEMENT

 001D C5           102         PUSH    B
 001E 47           103         MOV     B,A             ; SAVE DATA TO WRITE IN B-REGISTER
 001F 3E00         104         MVI     A,CLRACT        ; CLEAR WRITE COMPLETE & ILLEGAL ACCESS F/F'S
 0021 D322         105         OUT     CLRPRT          ; ACTIVATE CLEAR FUNCTION
 0023 3E03         106         MVI     A,CLRINA        ; DEACTIVATE CLEAR FUNCTION
 0025 D322         107         OUT     CLRPRT
 0027 3E0F         108         MVI     A,0FH           ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 0029 D3A0         109         OUT     EEPDR           ; OUTPUT TO PORT DIRECTION REGISTER
 002B 78           110         MOV     A,B             ; FETCH DATA TO WRITE
 002C D3A1         111         OUT     DATPRT          ; OUTPUT TO 2816 DATA LINES
 002E 7D           112         MOV     A,L             ; GET LOW ORDER ADDRES
 002F D3A2         113         OUT     ADRPRT          ; OUTPUT TO ADDRESS LINES
 0031 7C           114         MOV     A,H             ; GET HIGH ORDER ADDRESS
 0032 E607         115         ANI     7H              ; CLEAR ALL CONTROL LINES
 0034 F610         116         ORI     10H             ; ADD MUX BIT TO SELECT I/O PORTS FOR WRITE
 0036 D3A3         117         OUT     CTLPRT          ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 0038 F608         118         ORI     8H              ; ADD CE ACTIVE BIT
 003A D3A3         119         OUT     CTLPRT          ; OUTPUT CONTROL LINES AGAIN
 003C 47           120         MOV     B,A             ; SAVE HIGH ORDER ADDR/CTL LINE DATA
 003D 3EC0         121         MVI     A,COUNTL        ; OUTPUT TIMER COUNT (LOW ORDER)
 003F D3A4         122         OUT     TIMLOW
 0041 3E83         123         MVI     A,COUNTH        ; OUTPUT TIMER COUNT (HIGH ORDER)
 0043 D3A5         124         OUT     TIMHI
 0045 3ECF         125         MVI     A,0CFH          ; START THE TIMER
 0047 D3A0         126         OUT     EEPDR
 0049 78           127         MOV     A,B             ; RETRIEVE ADDRESS/CONTROL BITS
 004A F620         128         ORI     20H             ; ADD VPP ACTIVE BIT
 004C D3A3         129         OUT     CTLPRT          ; ACTIVATE VPP
 004E 320000   D   130         STA     TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
 0051 C1           131         POP     B               ; RESTORE REGISTERS
 0052 F1           132         POP     PSW
 0053 CD0000   E   133         CALL    WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
 0056 C9           134         RET                     ; AND RETURN BACK TO MAIN PROGRAM
                   135
                   136
                   137
                   138         DSEG
                   139
 0000              140 TEMCTL: DS      1               ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                   141
                   142
                   143
                   144         CSEG
                   145
                   146
                   147         ;       CONTROLLER IV END-OF-WRITE/ERASE-CYCLE ROUTINE
                   148
                   149         ;       CALLED TO BY I/O POLL OR INTERRUPT DRIVER AFTER WRITE COMPLETE
                   150         ;       TO SHUT DOWN CONTROLLER.
                   151
```

AFN-01885A

**Figure 25. Controller IV Software Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   4
  LOC  OBJ         LINE       SOURCE STATEMENT

                    152        ;       DATA PASSED:    TEMCTL (1 RAM BYTE) CONTAINING HIGH ORDER
                    153        ;                       ADDRESS (3 BITS) & CONTROL BEFORE WRITE COMPL.
                    154
                    155 ENDWE:
  0057 F5           156        PUSH    PSW             ; SAVE REGISTERS WE'LL DESTROY
  0058 D5           157        PUSH    D
  0059 3A0000   D   158        LDA     TEMCTL          ; GET ADDRESS LINES/CONTROL BITS
  005C E61F         159        ANI     1FH             ; REMOVE ACTIVE VPP BIT
  005E D3A3         160        OUT     CTLPRT          ; DE-ACTIVATE VPP
                    161
  0060 F5           162        PUSH    PSW             ; SAVE HIGH ORDER ADDRESS/CONTROL LINES
  0061 110D00       163        LXI     D,13D           ; SET UP COUNT FOR 100 USEC DELAY
                    164 DELAY:
  0064 1B           165        DCX     D               ; DELAY WHILE VPP FALLS
  0065 7A           166        MOV     A,D             ; DONE COUNTING?
  0066 B3           167        ORA     E
  0067 C26400   C   168        JNZ     DELAY           ; NO; KEEP LOOPING
                    169
  006A F1           170        POP     PSW             ; RESTORE ADDRESS/CONTROL LINES
  006B E617         171        ANI     17H             ; REMOVE CE ACTIVE BIT
  006D D3A3         172        OUT     CTLPRT          ; DE-ACTIVATE CE
  006F E607         173        ANI     7H              ; REMOVE MUX SELECT WRITE BIT
  0071 D3A3         174        OUT     CTLPRT          ; LET MUX SELECT FOR READ OPERATIONS
  0073 3E0E         175        MVI     A,0EH           ; PUT DATA PORT BACK TO INPUT MODE
  0075 D3A0         176        OUT     EEPDR           ; SO AS NOT TO CAUSE CONTENTION W/ DATA BUS
                    177
  0077 D1           178        POP     D               ; RESTORE REGISTERS
  0078 F1           179        POP     PSW
  0079 C9           180        RET                     ; AND EXIT
                    181
                    182
                    183
                    184        END


PUBLIC SYMBOLS
ENDWE  C 0057    READ   C 0000    WECYCL C 001C

EXTERNAL SYMBOLS
WEDELY E 0000

USER SYMBOLS
ADRPRT A 00A2    CLRACT A 0000    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083    COUNTL A 00C0    CTLPRT A 00A3
DATPRT A 00A1    DELAY  C 0064    EEPDR  A 00A0    ENDWE  C 0057    READ   C 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WECYCL C 001C    WEDELY E 0000

ASSEMBLY COMPLETE,  NO ERRORS                                                          AFN-01885A
```

**Figure 25. Controller IV Software Driver (Continued)**



AFN-01913A

```
ASM80 :F1:CCLR2.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE     SOURCE STATEMENT

                    1 $DEBUG
                    2
                    3
                    4         PUBLIC  CERASE
                    5
                    6         EXTRN   WEDELY,ENDWE
                    7
                    8         CSEG
                    9
                   10
                   11
                   12         ;       CONTROLLER II CHIP ERASE SUBROUTINE
                   13
                   14         ;       DATA PASSED:    NONE
                   15         ;       DATA RETURNED:  NONE
                   16         ;       REGS DESTROYED: NONE
                   17         ;       CALLS           WEDELY ( I/O POLL ROUTINE OR INTERRUPT DRIVER)
                   18
                   19         ;       I/O PORTS USED:
                   20         ;               PORT 22H (OUTPUT)
                   21         ;                       BIT 0 = WRITE COMPLETE CLEAR (ACTIVE LOW)
                   22         ;                       BIT 1 = ILLEGAL ACCESS CLEAR (ACTIVE LOW)
                   23         ;                       BIT 5 = CHIP CLR (+12V TO OE' LINE) (ACTIVE HI)
                   24
                   25         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                   26         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                   27         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                   28         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                   29         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                   30         ;                       CERASE.
                   31
                   32
                   33         ;       I/O SYMBOLS
                   34
  0022             35 CLRPRT  EQU     22H             ; CHIP ERASE OUTPUT PORT
  0000             36 CLRACT  EQU     00H             ; ACTIVE RESET OF CLEAR WC & ILL ACC FLIP-FLOPS
  0003             37 CLRINA  EQU     03H             ; INACTIVE RESET OF CLEAR WC & IA FUNCTION
  0023             38 CLRCCL  EQU     23H             ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                   39                                 ; OE' = +12V FUNCTION FOR CHIP CLEAR
                   40 CERASE:
  0000 F5          41         PUSH    PSW             ; SAVE REGISTERS
  0001 E5          42         PUSH    H
  0002 3E00        43         MVI     A,CLRACT        ; GET BITS TO RESET WRITE COMPL AND ILL ACC
  0004 D322        44         OUT     CLRPRT
  0006 3E23        45         MVI     A,CLRCCL        ; GET BITS TO DEACTIVATE CLEAR FUNCTION AND
                   46                                 ; TURN ON OE' = +12V FUNCTION FOR CHIP ERASE
  0008 D322        47         OUT     CLRPRT          ; OUTPUT TO I/O PORT
  000A 3EFF        48         MVI     A,0FFH          ; WRITE 0FFH TO THE 2816
  000C 3200A0      49         STA     0A000H                                          AFN-01885A
```

**Figure 26. Controller II Chip Erase Routines**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   2
  LOC  OBJ        LINE       SOURCE STATEMENT

  000F CD0000   E    50        CALL    WEDELY          ; GO TO I/O POLL LOOP OR INTERRUPT DRIVER
  0012 3E03          51        MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
  0014 D322          52        OUT     CLRPRT
  0016 E1            53        POP     H               ; RESTORE REGISTERS
  0017 F1            54        POP     PSW
  0018 C9            55        RET
                     56
                     57        END


PUBLIC SYMBOLS
CERASE C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000    WEDELY E 0000

USER SYMBOLS
CERASE C 0000    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    ENDWE  E 0000    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS                                                                    AFN-01885A
```

**Figure 26. Controller II Chip Erase Routines (Continued)**



AFN-01913A

```
ASM80 :F1:CCLR34.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE      SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4         CSEG
                    5
                    6
                    7         PUBLIC  CERASE
                    8
                    9         EXTRN   WEDELY,ENDWE
                   10
                   11
                   12         ;       CONTROLLER III I/O PORT DEFINITIONS
                   13
                   14         ;       IMPLEMENTED IN 8155 RAM / I/O  / TIMER CHIP
                   15
                   16
                   17         ;       PORT    DESCRIPTION
                   18         ;       ----    -----------
                   19         ;       0A0H    PORT DIRECTION REGISTER (SET TO 0FH = ALL PORTS OUTPUT)
                   20         ;
                   21         ;       0A1H    2816 DATA (OUTPUT)
                   22         ;
                   23         ;       0A2H    2816 LOW ORDER ADDRESS, A0-A7 (OUTPUT)
                   24         ;
                   25         ;       0A3H    2816 HIGH ORDER ADDRESS AND CONTROL LINES (OUTPUT)
                   26         ;                       BITS 0-2:       A8-A10
                   27         ;                       BIT  3:         CE CTRL (0=SELECT READ,
                   28         ;                                               WRITE ENABLE)
                   29         ;                       BIT 4:          MUX CTRL (0=READ,1=WRITE)
                   30         ;                       BIT 5:          VPP CTRL (0=INACTIVE,1=ACTIVE)
                   31         ;
                   32         ;       0A4H    LOW ORDER TIMER COUNT REGISTER
                   33         ;
                   34         ;       0A5H    HIGH ORDER TIMER COUNT REGISTER
                   35         ;
                   36         ;       22H     CHIP ERASE, INTERRUPT F/F CLEAR PORTS (OUTPUT)
                   37         ;                       BIT 0:          WRITE COMPL CLEAR (ACTIVE LOW)
                   38         ;                       BIT 1:          ILLEGAL ACC CLEAR (ACTIVE LOW)
                   39         ;                       BIT 5:          CHIP ERASE (+12V TO OE') ACT HI
                   40
                   41
                   42
 00A0              43 EEPDR   EQU     0A0H            ; PORT DIRECTION REGISTER
 00A1              44 DATPRT  EQU     0A1H            ; 2816 DATA (OUTPUT)
 00A2              45 ADRPRT  EQU     0A2H            ; 2816 LOW ORDER ADDRESS (OUTPUT)
 00A3              46 CTLPRT  EQU     0A3H            ; 2816 HIGH ORDER ADDRESS AND CONTROL (OUTPUT)
 00A4              47 TIMLOW  EQU     0A4H            ; LOW ORDER TIMER COUNT REGISTER
 00A5              48 TIMHI   EQU     0A5H            ; HIGH ORDER TIMER COUNT REGISTER
                   49
                                                                                        AFN-01885A
```

**Figure 27. Controller III, IV Chip Erase Routines**

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
 LOC  OBJ        LINE       SOURCE STATEMENT

 00C0             50 COUNTL  EQU     0C0H           ; LOW ORDER TIMER COUNT FOR 10 MSEC DELAY
 0083             51 COUNTH  EQU     83H            ; HIGH ORDER TIMER COUNT FOR 10 MSEC DELAY
                  52
                  53
                  54
                  55         ;       CONTROLLER III, IV CHIP ERASE SUBROUTINE
                  56
                  57         ;       DATA PASSED:    HL = ADDRESS OF 2816 LOCATION TO WRITE
                  58         ;                       A  = DATA TO WRITE
                  59         ;                            OR 0FFH  (ERASE)
                  60         ;       DATA RETURNED:  NONE
                  61         ;       REGS DESTROYED: NONE
                  62         ;       RAM REQUIRED:   1 BYTE FOR TEMP ADDRESS/CONTROL STORAGE
                  63         ;       CALLS:          PEDELY (I/O POLL ROUTINE OR INTERRUPT DRIVER)
                  64         ;
                  65         ;       COMMENTS:       ENDWE (END OF WRITE/ERASE CYCLE) ROUTINE
                  66         ;                       IS CALLED BY INTERRUPT DRIVER OR I/O POLL
                  67         ;                       ROUTINE (WEDELY) TO SHUT DOWN CONTROLLER.
                  68         ;                       THIS SUBROUTINE IS PART OF THE DRIVER
                  69         ;                       PACKAGE ROUTINES INITIATED BY A CALL TO
                  70         ;                       WECYCL.
                  71
                  72
 0000             73 CLRACT  EQU     0H             ; ACTIVE CLEAR WRITE COMPL & ILL ACC FUNCTION
 0023             74 CLRCCL  EQU     23H            ; DATA TO DEACTIVATE CLEAR WC & IA BUT ACTIVATE
                  75                                ; OE' = +12V FUNCTION FOR CHIP ERASE
 0003             76 CLRINA  EQU     3H             ; INACTIVE CLEAR WC & IA FUNCTION
 0022             77 CLRPRT  EQU     22H            ; PORT USED TO CLEAR ILL ACC & WRT COMPL F/F
                  78
                  79
                  80 CERASE:
 0000 F5          81         PUSH    PSW            ; SAVE REGISTERS
 0001 3E00        82         MVI     A,CLRACT       ; CLEAR WRITE COMPLETE AND ILL ACC FLIP-FLOPS
 0003 D322        83         OUT     CLRPRT
 0005 3E23        84         MVI     A,CLRCCL       ; DE-ACTIVATE CLEAR FUNCTION & SET OE' = +12V
 0007 D322        85         OUT     CLRPRT
 0009 3E0F        86         MVI     A,0FH          ; PUT ALL 8155 I/O PORTS IN OUTPUT MODE
 000B D3A0        87         OUT     EEPDR          ; OUTPUT TO PORT DIRECTION REGISTER
 000D 3EFF        88         MVI     A,0FFH         ; DATA TO WRITE IS ALL 1'S
 000F D3A1        89         OUT     DATPRT         ; OUTPUT TO 2816 DATA LINES
 0011 3E00        90         MVI     A,0            ; LOW ORDER ADDR (WE WRITE TO A000 FOR CCLR)
 0013 D3A2        91         OUT     ADRPRT         ; OUTPUT TO ADDRESS LINES
 0015 3E10        92         MVI     A,10H          ; ACTIVATE MUX FOR WRITE OPERATION
 0017 D3A3        93         OUT     CTLPRT         ; OUTPUT HIGH ORDER ADDRESS AND CONTROL LINES
 0019 F608        94         ORI     8H             ; ADD CE ACTIVE BIT
 001B D3A3        95         OUT     CTLPRT         ; OUTPUT CONTROL LINES AGAIN
 001D 3EC0        96         MVI     A,COUNTL       ; OUTPUT TIMER COUNT (LOW ORDER)
 001F D3A4        97         OUT     TIMLOW
 0021 3E83        98         MVI     A,COUNTH       ; OUTPUT TIMER COUNT (HIGH ORDER)
 0023 D3A5        99         OUT     TIMHI
 0025 3ECF       100         MVI     A,0CFH         ; START THE TIMER                    AFN-01885A
```

**Figure 27. Controller III, IV Chip Erase Routines (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    3
  LOC  OBJ         LINE        SOURCE STATEMENT

  0027 D3A0         101         OUT     EEPDR
  0029 3E38         102         MVI     A,38H           ; ACTIVATE VPP, CE' AND MUX
  002B D3A3         103         OUT     CTLPRT          ; ACTIVATE VPP
  002D 320000   D   104         STA     TEMCTL          ; SAVE HIGH ADDRESS/CONTROL BITS FOR AFTER INTR
  0030 CD0000   E   105         CALL    WEDELY          ; WAIT FOR END OF WRITE CYCLE BY I/O POLL OR
                    106                                 ; INTERRUPT DRIVER ROUTINE
  0033 3E03         107         MVI     A,CLRINA        ; DEACTIVATE CHIP CLEAR FUNCTION
  0035 D322         108         OUT     CLRPRT
  0037 F1           109         POP     PSW
  0038 C9           110         RET                     ; BACK TO CALLING ROUTINE
                    111
                    112
                    113
                    114         DSEG
                    115
  0000              116 TEMCTL: DS      1               ; RAM LOCATION FOR TEMP STORAGE OF CONTROL BITS
                    117
                    118
                    119
                    120         END


PUBLIC SYMBOLS
CERASE C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000    WEDELY E 0000

USER SYMBOLS
ADRPRT A 00A2    CERASE C 0000    CLRACT A 0000    CLRCCL A 0023    CLRINA A 0003    CLRPRT A 0022    COUNTH A 0083
COUNTL A 00C0    CTLPRT A 00A3    DATPRT A 00A1    EEPDR  A 00A0    ENDWE  E 0000    TEMCTL D 0000    TIMHI  A 00A5
TIMLOW A 00A4    WEDELY E 0000

ASSEMBLY COMPLETE,   NO ERRORS                                                              AFN-01885A
```

**Figure 27. Controller III, IV Chip Erase Routines (Continued)**



AFN-01913A

```
ASM80 :F1:IOPOLL.SRC
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1


 LOC  OBJ        LINE       SOURCE STATEMENT

                    1 $DEBUG
                    2
                    3
                    4
                    5
                    6        ;       ****************************************************
                    7
                    8        ;               2816 CONTROLLER I/O POLL ROUTINE
                    9
                   10        ;
                   11        ;
                   12
                   13        ;       ****************************************************
                   14
                   15
                   16        PUBLIC  PEDELY
                   17
                   18        EXTRN   ENDPE
                   19
                   20
                   21        CSEG
                   22
                   23
                   24        ;       PEDELY:  PROGRAM/ERASE CYCLE DELAY ROUTINE
                   25
                   26        ;                DELAYS VIA I/O POLLED WAIT LOOP ON 'WRITE COMPLETE'
                   27        ;                BIT.
                   28
                   29        ;       DATA PASSED:    NONE
                   30        ;       DATA RETURNED:  NONE
                   31        ;       REGS DESTROYED: NONE
                   32
                   33        ;       I/O PORT USED:  PORT 21H
                   34        ;                         - BIT 1 = 'WRITE COMPLETE' (ACTIVE HIGH)
                   35
                   36
 0021              37 WCPORT  EQU     21H             ; I/O PORT CONTAINING WRITE COMPLETE BIT
                   38
                   39
                   40 PEDELY:
 0000 F5           41         PUSH    PSW             ; SAVE A-REG, FLAGS
                   42 LOOP:
 0001 DB21         43         IN      WCPORT          ; GET WRITE COMPLETE BIT
 0003 E602         44         ANI     2H              ; MASK WC BIT
 0005 CA0100   C   45         JZ      LOOP            ; IF NOT SET THEN KEEP WAITING
                   46
 0008 F1           47         POP     PSW             ; RESTORE A, FLAGS                    AFN-01885A
```

**Figure 28. Controller I/O Poll Routines**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2
  LOC             LINE        SOURCE STATEMENT

  0009 CD0000   E    48         CALL    ENDPE          ; CALL END PROGRAM/ERASE CYCLE ROUTINE TO
                     49                                ; SHUT DOWN 2816.
  000C C9            50         RET                    ; RETURN BACK TO HOST PROGRAM.
                     51
                     52
                     53
                     54         END


PUBLIC SYMBOLS
PEDELY C 0000

EXTERNAL SYMBOLS
ENDPE  E 0000

USER SYMBOLS
ENDPE  E 0000    LOOP   C 0001    PEDELY C 0000    WCPORT A 0021

ASSEMBLY COMPLETE,  NO ERRORS                                          AFN-01885A
```

**Figure 28. Controller I/O Poll Routines (Continued)**

```
ASM80 :F1:INTER.SRC MOD85
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   1

 LOC  OBJ        LINE      SOURCE STATEMENT
                    1 $DEBUG
                    2
                    3
                    4
                    5         CSEG
                    6
                    7         PUBLIC  WEDELY,HANDLE
                    8
                    9         EXTRN   ENDWE
                   10
                   11         CSEG
                   12
                   13
                   14
                   15         ;       WEDELY - WRITE/ERASE CYCLE DELAY SUBROUTINE
                   16         ;              - INTERRUPT DRIVEN
                   17
                   18         ;       CALLED TO WAIT FOR INTERRUPT TO OCCUR WHILE WAITING OUT
                   19         ;       2816 CONTROLLER WRITE CYCLE
                   20
                   21         ;       DATA PASSED:    NONE
                   22         ;       REGS DESTROYED: NONE
                   23         ;       INTERRUPT USED: EXPECTS CONTROLLER TO USE INTERRUPT 6.5
                   24         ;                       MASKS OUT ALL OTHER INTERRUPTS
                   25         ;       USED WITH:      INTERRUPT HANDLER SUBOURINE 'HANDLE'
                   26
                   27         ;       RAM REQD:       1 BYTE - 'WRTCOM' - WRITE COMPLETE INTERCOM
                   28         ;                              - BIT ZERO SET BY INTERRUPT HANDLER
                   29         ;                                WHEN WRITE COMPLETE.
                   30
                   31
  000D             32 IONMSK  EQU     1101B          ; INTERRUPT MASK ENABLING INTERRUPT 6.5 ONLY
                   33
                   34 WEDELY:
  0000 F5          35         PUSH    PSW            ; SAVE A-REGISTER, FLAGS
  0001 AF          36         XRA     A              ; ZERO WRITE COMPLETE INTERCOM REGISTER
  0002 320000    D 37         STA     WRTCOM
  0005 3E0D        38         MVI     A,IONMSK       ; ENABLE INTERRUPT 6.5 ONLY
  0007 30          39         SIM
  0008 FB          40         EI                     ; ALLOW INTERRUPTS TO OCCUR
                   41 LOOP:
  0009 3A0000    D 42         LDA     WRTCOM         ; GET WRITE COMPLETE STATUS REGISTER
  000C 1F          43         RAR                    ; PUT LEAST SIGNIFICANT BIT INTO CARRY
  000D D20900    C 44         JNC     LOOP           ; IF LSB NOT SET, KEEP LOOPING
                   45
  0010 F1          46         POP     PSW            ; RESTORE A-REGISTER
  0011 C9          47         RET                    ; BACK TO HOST PROGRAM
                   48
                   49
                   50
```

AFN-01885A

**Figure 29. Controller Interrupt Driver**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE  PAGE   2
 LOC  OBJ        LINE       SOURCE STATEMENT

                   51
                   52          DSEG                ; SAVE A RAM LOCATION
 0000              53 WRTCOM:  DS      1
                   54
                   55
                   56
                   57
                   58          ASEG
                   59
 0FE0              60          ORG     0FE0H
                   61
                   62
                   63
                   64          ;       HANDLE - 2816 CONTROLLER INTERRUPT HANDLER
                   65          ;                UPON RECEIPT OF INTERRUPT, WRITE COMPLETE BIT CHECKED.
                   66          ;                IF SET, 'ENDWE' IS CALLED TO SHUT DOWN CONTROLLER.
                   67          ;                IF ILLEGAL ACCESS BIT SET, 'ILLACC' IS JUMPED TO.
                   68          ;                IF NEITHER BIT SET, 'BADINT' IS JUMPED TO INDICATING
                   69          ;                BAD INTERRUPT OCCURED.
                   70
                   71          ;       DATA PASSED:    NONE
                   72          ;       REGS AFFECTED:  NONE
                   73          ;       REQUIRES:       HOST PROGRAM MUST SET UP INTERRUPT VECTOR
                   74          ;                       SO 'HANDLE' EXECUTED UPON RECEIPT OF RST 6.5
                   75          ;                       COMMAND.
                   76          ;       CODE REQUIRED:  'ENDWE' SUBROUTINE CALLED TO SHUT DOWN
                   77          ;                       CONTROLLER AT END OF PROGRAM/ERASE CYCLE
                   78          ;       RAM USED:       1 BYTE - WRTCOM - WRITE COMPLETE STATUS BYTE
                   79          ;                                 - BIT 0 SET WHEN WRITE COMPLETE
                   80          ;
                   81          ;       I/O PORT USED:  PORT 21:
                   82          ;                         - BIT 0 = WRITE COMPLETE (ACTIVE HI)
                   83          ;                         - BIT 1 = ILLEGAL ACCESS (ACTIVE HI)
                   84
                   85
 000A              86 IOFMSK   EQU     1010B           ; MASK OUT INTERRUPT 6.5
 0021              87 WCPORT   EQU     21H             ; WRITE COMPLETE STATUS I/O PORT
                   88
                   89 HANDLE:
 0FE0 F5           90          PUSH    PSW             ; SAVE A-REG, FLAGS
 0FE1 DB21         91          IN      WCPORT          ; PICK UP CONTROLLER STATUS BITS
 0FE3 1F           92          RAR                     ; MOVE ILLEGAL ACCESS BIT INTO CARRY
 0FE4 DA1200    C  93          JC      ILLACC          ; GO TO ILLEGAL ACCESS ROUTINE IF BIT SET
 0FE7 1F           94          RAR                     ; MOVE WRITE COMPLETE BIT INTO CARRY
 0FE8 D21300    C  95          JNC     BADINT          ; IF NOT SET THEN GO TO BAD INTERRUPT HANDLER
 0FEB 3E0A         96          MVI     A,IOFMSK        ; UN-MASK 6.5 INTERRUPTS
 0FED 30           97          SIM
 0FEE CD0000    E  98          CALL    ENDWE           ; SHUT DOWN CONTROLLER
 0FF1 3E01         99          MVI     A,1H            ; SET WRITE COMPLETE INTERCOM BIT
 0FF3 320000    D 100          STA     WRTCOM          ; AND SAVE IN RAM                    AFN-01885A
```

**Figure 29. Controller Interrupt Driver (Continued)**



AFN-01913A

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   3
  LOC  OBJ        LINE        SOURCE STATEMENT

  0FF6 F1          101         POP     PSW             ; RESTORE REGISTER
  0FF7 C9          102         RET                     ; AND RETURN BACK TO INTERRUPTED ROUTINE
                   103
                   104
                   105
                   106
                   107
                   108
                   109
                   110         CSEG
                   111
                   112
  0012 C7          113 ILLACC: RST     0               ; ILL ACCESS RESTART VECTOR FOR TESTING ONLY
  0013 C7          114 BADINT: RST     0               ; BAD INTERRUPT RESTART VECTOR FOR TESTING ONLY
                   115
                   116         END


PUBLIC SYMBOLS
HANDLE A 0FE0    WEDELY C 0000

EXTERNAL SYMBOLS
ENDWE  E 0000

USER SYMBOLS
BADINT C 0013    ENDWE  E 0000    HANDLE A 0FE0    ILLACC C 0012    IOFMSK A 000A    IONMSK A 000D    LOOP   C 0009
WCPORT A 0021    WEDELY C 0000    WRTCOM D 0000
ASSEMBLY COMPLETE,   NO ERRORS                                                              AFN-01885A
```

**Figure 29. Controller Interrupt Driver (Continued)**



AFN-01913A

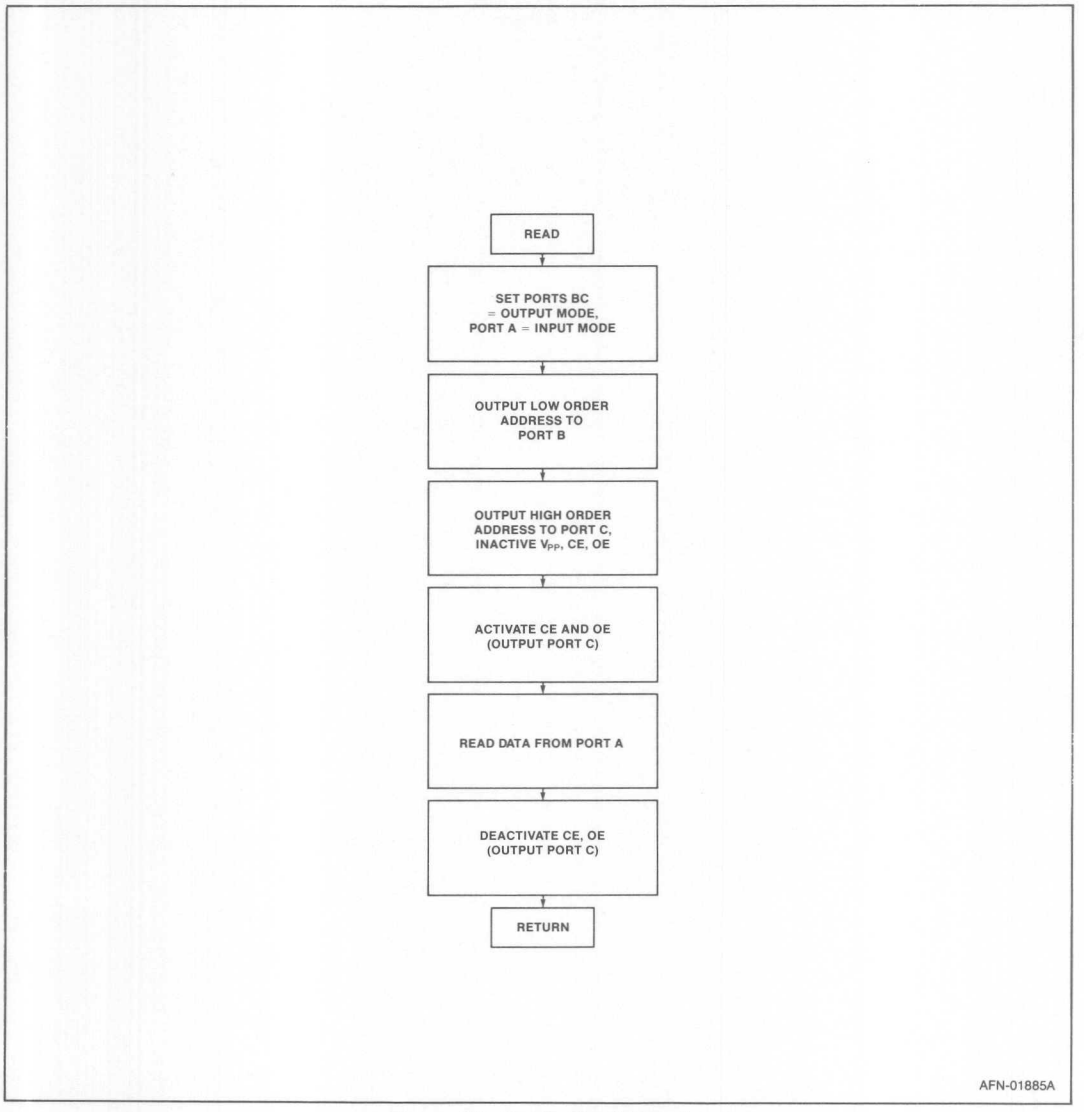



**Figure 30. Controller IV Read**